TYPOLOGY OF WORD AND AUTOMATIC WORD SEGMENTATION IN URDU TEXT CORPUS

MS Thesis

Submitted in Partial Fulfillment
Of the Requirements of the
Degree of

Master of Science (Computer Science)

AT
NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES
LAHORE, PAKISTAN
DEPARTMENT OF COMPUTER SCIENCE

By
Nadir Durrani
August 2007

Approved:

____

Head
(Department of Computer
Science)

Approved by Committee Members:

Advisor

____

Dr. Sarmad Hussain
Professor
FAST - National University

Other Members:

____

Mr. Shafiq-ur-Rahman
Associate Professor
FAST - National University

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance of a thesis entitled “Typology of Word and Automatic Word Segmentation in Urdu Text Corpus" by Nadir Durrani in partial fulfillment of the requirements for the degree of Master of Science.

Dated: August 2007

To my parents

Vita

Mr. Nadir Durrani received a Bachelor of Science degree in Computer Science from National University of Computer and Emerging Science (NUCES), Lahore in 2004. Nadir Durrani has been a member of CRULP since 2003. He has been working as a Research Officer in different R&D works. The research in this dissertation was carried out from 2006 to 2007.

Acknowledgements

First of all I would like to thank the Almighty who gave me strength and path to pave my way during the course of this research.

I would like to thank Dr. Sarmad who changed my mind and brought me back to the field of Computer Science and put me back on track. His continuous guidance and supervision of this work has helped me as a torch to accomplish this task.

My utmost thanks to Mr. Shafique Ur Rehman for not putting hard questions during proposal and final defense and letting me go easy.

My special thanks to Madiha Ijaz for providing all the data that I ever wanted to engineer this problem. She contributed a lot by providing all sorts of data that I needed.

Special thanks to Nayyara and Tahira for their discussions on what is word and for marking word boundaries on survey.

To my friend Hasaan for daily discussion on what is word and not getting anything out of it, also for providing me with smoothed bigram frequencies.

To Ahmad Muaz for his translations.

To all those who filled the survey form and marked word boundaries on those stupid sentences.

And last and the most important person my mother who made me whatever I am today. Her prayers and continous encouragement have led me to this point.

Nadir Durrani

Contents

1. Introduction

Word segmentation is the foremost obligatory task in all NLP application. The initial phase of text analysis for any language processing task usually involves tokenization of the input into words. Be it a text to speech system (TTS), machine translation, spell & grammar checking, information retrieval or a part of speech (POS) tagger. A TTS must have the information about word boundaries to be able to articulate properly and to place pause and stress appropriately. The syntactic and semantic analysis in machine translation systems are based on words and the neighboring ones. A spell checker requires word boundary information for error word based on which it could suggest list of possible corrections. Similarly, a rule based POS tagger needs preceding and following few words and their POS tags to properly tag current word.

For inflectional languages like English, French and Dutch etc. tokenization is considered trivial because the white space or punctuation marks between words is a good approximation of where a word boundary is. Whilst in various Asian languages, white spaces is rarely or never used to determine the word boundaries, so one must resort to higher levels of information such as: lexicon, information of morphology, syntax and even semantics and pragmatics to reconstruct the word boundary information.

Urdu is amongst the Asian languages that suffer word segmentation dilemma. However, unlike other Asian languages word-segmentation in Urdu is not just a space insertion problem. Space in Urdu is a frequently used character in printed texts. However its presence does not necessarily indicate word boundary. In other cases space is optionally used so the user enjoys liberty. Put it another way a sentence can have cases when a single word might have space in between. Alternatively multiple words are written in continuum without any space like in CJK languages. So, Urdu word-segmentation is both a space insertion and a space deletion problem. To further complicate the situation some words that are written with space can be also be written without them. In few cases these are spelled differently when written without space.

Urdu word segmentation problem is triggered by its orthographic rules and confusion about the definition of word. There is no consensus on what exactly is a word in Urdu. In the books of grammar authors tend to ignore this issue or try to define it rather abstractly. For example [1] has defined word like this:

- ‘لفظ :- انسان منہ سے بولتے وقت جو کچھ نکالتا ہے اسے لفظ کہتے ہیں ۔’
    - ‘Word: Whatever comes out your mouth while speaking is known as word.’

Primary class books define word as:

- 'لفظ :- جملے کے بامعنی حصوں میں سے ہر حصے کو لفظ کہتے ہیں ۔ '
    - ‘Word: Each constituent in the meaningful constituents of a sentence is a word.’
- 'لفظ :- ایک سے زیادہ حروف کے مرکب کو لفظ کہتے ہیں ۔'
    - ‘Word: More than one letters combine together to form a cluster known as a word.’

All of these fuzzily define what an Urdu word is but none of these is concrete enough to be used as a yard-stick. In other words given a sentence and these definitions different users would still identify different sets of words. Books written on Urdu grammar [2] & [3] by foreign authors do not define word at all and start with POS straight away.

However we can not ignore or abstractly address this issue. Before formulating a solution we first need to clearly model our own definition of word. This thesis comprises of six sections. The first section provides a literature review on linguistic and orthographic background of problem. Second section discusses the problem in detail and the words that cause segmentation problems in Urdu. Third section draws a conclusion, defines word, the definition that we will use during the course of this work.

Section four provides literature review on existing techniques to solve word segmentation problem in other Asian languages. Section five defines methodology and the algorithm employed to solve this problem. Section six provides results and discusses future work with concluding remarks.

2. Linguistic Literature Review and Background

2.1. What is a Word?

There are several different notions of word. Most of the textbooks commonly talk about the following three[1]:

- Orthographic Word
- Phonological Word
- Lexical Word

2.1.1. Orthographic Word

An orthographic word is one in which word boundaries are defined by some orthographic rule. In English and some other languages space is frequently used to indicate word boundaries. For example last sentence has following word: ‘In’, ‘English’, ‘and’, ‘some’ and so on. Oscan, an ancient language of Italy uses dots to mark word boundaries. Consider text below:

STATUS•PUS•SET•HURTIN•KERRIIIN•VEZKEI•STATIF•

Figure 1: Oscan Language used Dots [41]

The idea of orthographic words has no existence in speech and is only important for written word. Although space is commonly used (in Latin based languages) to mark word boundaries there are few exceptions. Consider *green house*, *pocket knife*, *ice cream* and *common sense* etc. Each of these is two orthographic words but a native speaker may want to treat these as a single entity. These are all single words and are alternatively written as *greenhouse*, *pocketknife* and *commonsense*. In this case it qualifies our definition of being a single word.

Such variations are by no means rare in English. The word *landowner* can also be written as *land-owner* and *land owner*. All of these three are single word but our definition for orthographic recognizes fails to recognize them as single word. The rules for English do not specify which compounds to be written with space and which ones to be written without them. Moreover individual preferences vary considerably.

In South East Asian languages space is used only to mark sentence boundaries. So the orthographic word definition will totally fail in this scenario. Given is one example from Lao [18]:

[1] Most of this section is reported from [41]

ສຽງປົກກະຕິກໍຈິງຢູ່ແລ້ວສຽງຟ້າຮ້ອງ

Figure 2: Lao Language

These are three words ‘ສຽງປົກກະຕິ’, ‘ກໍຈິງຢູ່ແລ້ວ’ and ‘ສຽງຟ້າຮ້ອງ’ written in a continuum. Therefore it can be seen that orthographic word is of no use with such writing system. The notion of orthographic word is of very little interest in linguistic study. It might be important in study of writing systems but for linguistic purposes orthographic words are irrelevant.

2.1.2. Phonological Words

A phonological word is a piece of speech which acts as a unit of pronunciation. This is based on certain criteria which vary from language to language. In English each phonological word has exactly one main stress. Consider following sentence:

The rest of the books’ll have to go here.

Figure 3: Phonological Word

There are five main stress falling on [The rest], [of the books’ll], [have to], [go], [here]. So there are five phonological words [41].

In Hebrew last syllable is regularly stressed so the word boundary is likely to fall after each stressed syllable. In Turkish another phonological criteria known as vowel harmony exists. Vowel within a word share same quality. So a word boundary is like to occur when the quality changes. For example suffix meaning ‘in’ appears as ‘de’ or ‘da’ occurs depending upon previous morpheme. ‘In the house’ is ‘evde’ but ‘in the room’ is ‘odada’. Suffix ‘my’ has four forms ‘ev-im’ for ‘my house’, ‘at-Im’ for ‘my horse’, ‘gözüm’ for ‘my eye’ and finally ‘topum’ for ‘my gun’ [41].

Phonological words are important for the study of speech but they are irrelevant to the study of grammar.

2.1.3. Lexical Words

Lexical word is a unit of vocabulary. A lexical item is a word in a sense that how many words are there in English vocabulary and saying that I learned 20 new words of Urdu today. Lexemes are entries in dictionaries.

Lexemes are abstract forms that are represented in speech or writing by one of its possible several forms that it can take for grammatical purposes. For example ‘الماری’, ‘الماریاں’ and ‘الماریوں’ are three possible forms of lexical item ‘الماری’ in Urdu. Similarly lexical item *take* can be represented by any of the five grammatical forms *take*, *took*, *taken*, *taking* and *takes*. The dictionary only provides entry for one of the representation forms. The lexical item *take* only has a single entry which is represented by any of its five forms. Similarly ‘الماری’ has a single entry represented by ‘الماری’.

Some dictionaries might give more than one entries for one Lexical item but these are purely cross-reference to main-entry. So entry *took* in dictionary would cross-reference saying please see *take*.

2.1.3.1. Grammatical Words

Lexical entries contain content words. These are open class words and have semantic content and readily identifiable meaning. On contrary grammatical words are closed class words. They are known as empty word and have little identifiable meaning but has more grammatical function. It is easier to translate the content words in sentences because the equivalent always exists in corresponding dictionary. However grammatical word can not be easily translated because there might not be an equivalent. For example English phase 'bottle of wine' has corresponding Basque translation '*boteila bat ardo*' which means 'bottle a wine'. There is corresponding equivalent of 'of'. Similarly its corresponding Welsh translation is 'potel gwin' which means 'bottle wine' with no equivalent of 'of' and 'a' [41].

It is easier to coin open words. Closed class words can not be formed through word coinage process.

2.2. Compound Words

Compounding is one of the morphological processes of word-formation. When defined plainly a compound word is a combination of two already existing words [12]. According to [8] compound word is formed by concatenating root morphemes to form new stems. Similar definition is given by [9] according to which a compound word involves combining two or more lexemes' stems to form another lexeme.

Compounding is a process of forming new units of thought. It can be a single word with attached syllables (affixes, verb endings), composed of two or more words joined together plainly, by a hyphen or a binding morpheme or can be a group of words that express a single idea [15].

2.2.1. Compound Classes

When words combine the idea generated might be related to its constituents or can be entirely a new one. Depending upon how tightly a newly formed word is related to its constituents, compounds can be divided into following categories [12].

- Endocentric
- Exocentric
- Coordinative

2.2.1.1. Endocentric Compounds

In endocentric compound the head morpheme processes the basic meaning of the whole compound. Other morphemes act as modifier which acts to restrict this meaning [12]. In general the meaning of a compound is a specialization of head word. The part of speech of the compound is same as that of its head. The endocentric compounds themselves are divided into two categories [10].

- Descriptive
- Determinative

In ***descriptive compounds*** the modifier morpheme is used in attributive, appositional or adverbial manner. For example *Blackboard* in English is particular kind of word which is generally black. Another example is of Sanskrit 'Maharaja' means a king who is great. So a descriptive compound YX is formed by combining root morpheme 'X' and a modifier 'Y' such that 'Y' is a particular kind of 'X'.

On the other hand in ***determinative compounds*** ‘Y’ is not an attribute to ‘X’ it is rather related to ‘X’ in a way corresponding to one of the grammatical cases of ‘X’. In English it serves the same purpose as prepositions do. For example ‘Doghouse’ is the house where dog lives. ‘Raincoat’ means a coat against rain. Another example is of Sanskrit: ‘Raja-putra’ means son of king and yet another example is of Urdu ‘Namaz-e-Janaza’ means Janazay ki namaz (funeral prayer) in which ‘ki’ is a grammatical case that exhibits possessor relation. The two roots are connected with combining morphemes – e – in this case which is known as zer-e-izafat. More is discussed later in section Urdu Compounds.

2.2.1.2. Exocentric Compounds

Unlike endocentric compounds, the meaning of exocentric compounds does not follow meaning of its constituents [12]. Exocentric compounds are headless and their meanings can not be transparently determined by its parts. For example *white-collar* is not a kind of collar neither it is a white thing; it rather means ‘professional’ i.e. is a person with white collar. Similarly *open-minded* is a not a mind that is open it is rather a person who has an open-mind.

Word class of exocentric compounds is determined lexically regardless of the POS of its constituents. For example *must-have* is a noun and not a verb. A Sanskrit example of exocentric compounds is ‘Bahuvrihi’ which by part means ‘Bahu’ (much) and ‘virihi’ (rice) but actually means ‘rich man’ [10]. Exocentric compounds XY can be generally defined as a person/thing having ‘Y’ and the relation of ‘X’ to ‘Y’ is unspecified. Exocentric compounds occur more often in adjectives than nouns.

2.2.1.3. Coordinating Compounds

Coordinating compounds also known as copulatives or dvandva [10]. It refers to two or more morphemes that can be connected in a sense by conjunctions ‘and’. Copulatives usually combine nouns with similar meanings and the meaning of compound is a specialization. For example fighter-bomber represents an air-craft that is both fighter and a bomber. Other such example is of activist-scholar. An Urdu example of coordinating compounds is ‘Hajj-wa-Umrah[2]’ means ‘Hajj’ and ‘Umrah’ where ‘wa’ represents linking morpheme.

2.2.2. Compound Formation

Compound formation varies across different languages.

English is an analytical language; compounds are formed by conjoining words without case markers. Words can combined together to form infinitely long compounds. For example:

tube
feed-tube
in-line feed tube
in-line feed tube adaptor
in-line feed tube adaptor hose
in-line feed tube adaptor hose cover
in-line feed tube adaptor hose cover cleaning
in-line feed tube adaptor hose cover cleaning instruction
in-line feed tube adaptor hose cover cleaning instruction sheet

Figure 4: Compound Formation is Productive in English [12]

[2] Hajj and Umrah are rituals of Islam

However short compounds in English mostly contain 2-3 words and are written in following three ways:

In ***Solid*** or closed compounds 'X' and 'Y' have no space in between. These are mostly monosyllabic units and have been around in English for quite some time. *Housewife* and *background* are a few examples to name. Compound with more than two words can not be written in this way. So a *coffeepot cleaner* can be written as *coffee pot cleaner* or *coffee-pot cleaner* but never as **coffeepotcleaner* [9].

Open or spaced compounds as the name suggests is written with space between 'X' and 'Y'. *Ice cream* and *lawn tennis* are few examples.

Dashed or Hyphenated compounds would have X–Y structure. Example: *acetic-acid solution*. However, there is no hard and fast rule as to which words are compounded with hyphens and which with or without space. This is depends upon individual preferences. One may find *landlord*, *land-lord* and *land lord* within same text.

Chinese is another example of analytical language which also form compounds with nouns and no markers at all. For example *Hànyǔ* (漢語; simplified: 汉语), or "the Han Chinese [10].

In more synthetic languages like ***German*** compounds are formed by combining words that are case-marked. For example constituents of *Kapitänspatent* are *Kapitän* (*sea captain*) and *Patent* (*license*). These are joined by the genitive case marker *-s*. German language does not prohibit writing multi-stemmed compounds as single orthographic word. Consider following example [9].

Lenensversicherungsgesellschaftsangestellter
Lenen+s+versicherung+s+gesellschaft+s+angestellter
(Life+CompAug+insurance+CompAug+company+CompAug+employee_
'Life insurance company emplpoyee'

Figure 5: Compounding in German Language

Dutch compounding consists one of the following form of XY, X-en-Y, X-e-Y or X-s-Y. Which morpheme a particular compound is largely lexicalized and learned by Dutch speakers [9].

Compounding is a very productive phenomenon in **Russian**. Compound nouns can be agglutinative. In these agglutinative affix is used in these. Example of this is 'parokhod' (steamship): par + o + khod. Russian also has hyphen separated compounds e.g. (stol-kniga which means folded table). It also has abbreviated compounds like "Akademgorodok" (from "akademichesky gorodok", i.e. "Academic Village) [10].

Some language incorporates nouns into the verb when forming compounds. Consider example of Iroquin language **Onondaga**:

Pet wa? + ha +HTU + ?t +a? ne? o + **hwist** + a?
'Pat lost the money'

Pet wa ? + ha +hwist +AHTU + ?t +a?
'Pat lost money'

Figure 6: Compounding in Onondaga Language

In first example object is part of its noun; however it is integrated into verb and appears adjacent to the verb stem [9].

2.2.3. Types of Compounds

Nominal compounds are very frequently found in **English**. As mentioned already most formations follow the pattern modifier + head. One of the problems therefore is to differentiate between compounds and phases. For example compound it is hard to distinguish between compound 'hand gun' and phrasal construction 'tropical storm' because 'hand gun' can also be perceived as 'gun' as a head where modifier 'hand' tells what kind of 'gun' it is. Stress is one criterion that can be used to differentiate between these two. Compounds (e.g. GUNman, HANDwriting) usually have stress on their first element on contrary phases (personal HISTORY, human SUFFERING) have stress on their second element. This criteria, however is foolproof by no means because counter examples like because many people say APPLE cake but apple PIE although apple PIE is also a compound [9].

As already mentioned afore compounds can be formed by combining words in chains as many as possible, these can be constructed recursively by combining two words at a time. Consider example of Science fiction writer. These can be obtained by combining science and fiction and then writer to the resulting compound.

In **German** it is easy to distinguish between phases and compounds because of morphological markers which are obligatorily used on phrasal modifiers. Consider following two examples [9]:

Rot + er Wein
(Red + NOM/SG/MASC/STRONG wine)
'Wine which is red in color'

Rot + Wein
Red + Wine
'Red wine qua category of wine'

Figure 7: Phrases and Compounds in German

The enlection '-er' is mandatory required in phrasal constructions. Where as it is absent in compounds.

Nominal compounds in **French** have heads on left hand side. Moreover prepositional components are inserted before the modifiers. Examples are *chemin-de-fer* ("railway", lit. "road of iron") and *moulin à vent* ("windmill", lit. "mill (that works)-by-means-of wind") [10].

Verb-Noun Compound formation is very common in Indo-European languages. The verb and its object combine and convert simple verbal clause into a noun [10].

In **Spanish**, for example, such compounds consist of a verb used for third person singular, present tense, indicative mood followed by a noun (usually plural): e.g., *rascacielos* (modelled on "skyscraper", lit. "Scratches skies"), *sacacorchos* ("corkscrew", lit. "Removes corks") [9].

French and **Italian** have these same compounds with the noun in the singular form: Italian *grattacielo* ("skyscraper"), French *grille-pain* ("toaster", lit. "toasts bread") [10].

In **English** generally verb and noun both are in uninflected forms. Examples are *spoilsport*, *killjoy*, *spendthrift*, *cutthroat*, and *know-nothing.*

Also common in English is another type of verb-noun compounds, in which an argument of the verb is incorporated into the verb, which is then usually turned into a gerund, such as *breastfeeding*, *finger-pointing*, etc. The noun is usually an instrumental complement [10].

Compound Adjectives

In **English** the construction is similar to nominal compound construction. The right most element in a compound is its head. It is preceded by one or more modifiers. The function of a modifier is to restrict the head word [10].

Compound Ad positions

These are also frequently formed in English. These are formed by prepositions and nouns. *On top of* and *make up* are few examples to name. Similar example is of Spanish encima de. Similar pattern is observed in Japanese except that it uses postpositions instead. For instance: no naka which means 'on the inside of' [10].

2.3. Compound Words in Urdu

In Urdu, Compounding is a very rich phenomenon. Urdu is an off-shoot from many other languages like Arabic, Farsi, Turkish, Hindi and Sanskrit etc. Compounding is frequently seen to occur in Farsi and is inherited by Urdu as well [1].

Urdu, like English is a head final language. Compounds can be formed with two independent words such as noun and adjectives, and also with independent words and verb stems and verb stems themselves [3]. Compounds usually occur in following formats XY, X-o-Y and X-e-Y.

2.3.1. XY Formation

The XY formation simply involves combining two free-morphemes. No more than two morphemes can combine together in this manner in Urdu. Example of such formation is 'موم بتی' (MomBatti) which means Candle. Another example is 'جرائم پیشہ' (Jaraim Paisha) which means criminal.

According to [13] compounds in Urdu can be classified into four types:

2.3.1.1. Dvanda

These have two conditions:

- Both morphemes that form compound have different meanings. These further have two conditions:
  - Both morphemes are nouns. Example 'ماں باپ' (Maan Baap; Parents), 'ناک نقشہ' (Naak Naqsha; Features).
  - Both morphemes are verbs. Example 'پڑھا لکھا'(Parha Likha; Educated).

- Both morphemes that form compounds have identical or similar meanings. These also have further two conditions:
  - Both morphemes are nouns. Example 'خط پتر'(Khat Patar; Letter), 'کام کاج' (Kaam Kaaj; Work).
  - Both morphemes are verbs. Examples 'دیکھ بھال' (Daikh Bhal; Care taking).

2.3.1.2. Tatpurusa

This is another type of XY compounds in which means a type of Y which is related to X in a way corresponding to one of the grammatical cases of X. Examples are:

گھوڑ دوڑ (Ghur Dour; Horse Race)
چادر چھپول(Chadar Chapol)
دیس نکالا (Dais nikala; Exiled)

Figure 8: Tatpurusa Compounds

2.3.1.3. Karmadharaya

In this type of XY compounding the relation of first to second element is attributive, appositional or adverbial. These are often classified as sub-type of Tutpurusa. Examples are:

بڑکنّا (Khar Kanna; The one with big ears)
بڑھ بولا (Barh Bola; The one who exaggerates)

Figure 9: Karmadharya Compounds

2.3.1.4. Divigu

It is a type of XY formation in which X is a numeral. Examples are:

ادھ موا (Adh Muwa; Half Dead)
ڈوپٹّا (Dupatta; Scarf)

Figure 10: Divigu Compounds

2.3.2. X-o-Y Formation

The X-o-Y construction contains linking morpheme -o-. It usually gives mean of ‘and’ and is commonly used. Example are ‘ملک و ملت’ (Mulk-o-Milat; Country and Nation), ‘عزیزواقارب’ (Aziz-o-aqarib; near and dear ones).

The X-o-Y formation is an instance of coordinating compounds. The morpheme -o- is mostly involved in nominal constructions. There are cases when both morphemes in compounds give identical or similar meaning. For example in compound ‘تباہ و برباد’ (Tabah-o-barbad) both ‘تباہ’ (Tabah) and ‘برباد’ (Barbad) means destroy. Compound itself is used to give meaning of destroy and itself can be replaced by any of its constituents in a sentence to give exactly the same meaning. Similar another example is ‘امن و امان’ (Amn-o-aman) which means peace.

Although it is originated from Farsi the -o- is nowadays also used to combine English words. One such example is ‘پٹرول و ڈیزل’ (Petrol-wa-Diesel). Such examples are commonly found in Urdu corpra. The -o- is also used to form compounds having verb stems. These are discussed below.

2.3.3. X-e-Y Formation

The third and final formation X-e-Y contains linking morpheme or an enclitic short vowel known as zer-izafat or hamza-e-izafat. Izafat means increase or addition. It is pronounced in Urdu as short /e/ and is used in Noun-e-Noun and Noun-e-Adjective compounds.

The ***noun-e-noun*** compounds signify possessor relationship in which X belongs to Y [3]. Alternative construction for such compounds is Y 'کے/کی' (ke/ki) X where ke and ki are case-markers used to mark possession. Examples are 'اہلیان کراچی' (Ehliyan-e-Karachi) which means 'People of Karachi' and can be alternatively as 'کراچی کے لوگ' (Karachi ke log ; lit. Karachi of People; 'People of Karachi').

The ***noun-e-adjective*** formation shows that noun X is modified by adjective Y. For example: ' وزیر اعظم' (Vazeer-e-azam) which means prime minister. Another example is 'دیوان خاص' (Deewan-e-khas) which means private hall of audience [3]. These compounds however are lexical entries for native Urdu speakers.

Zer-e-Izafat is left unwritten in modern texts but a native speaker would pronounce it as if it is there. When written it is written as follow [3]:

- As subscript zer
- As hamza over bari yeh (when it follows word ending in the long vowels alef or vao)
- As hamza over choti heh (when it follows a final heh)
- As zero (when it follows word ending with bari yah)

2.4. Reduplication

Reduplication is a morphological process that involves repletion of part or all of a root. Reduplication may be full or partial [8]. In Urdu both forms of reduplication exists. It is normally used to put emphasis. Words are repeated to express multiplicity or variety [1]. Examples are given below:

Table 1: Pure Reduplication

Pronouns	Verbs
کوئ کوئ (few) کچھ کچھ (somewhat) کون کون (What various people)	گن گن (Count carefully) بہا بہا (Shedding) بدل بدل (Tossing and turning)
Nouns	**Adverbs**
گلی گلی (Every street) پتّا پتّا (Every leaf) بوٹا بوٹا (Every plant)	گھڑی گھڑی (Constantly) جہاں جہاں (Wherever) کبھی کبھی (Sometimes)

In some cases reduplication involves linking morphemes like /ma/, /ba/ or /a/ to form patterns like X-ma-X, X-ba-X or X-a-X respectively. Some examples are given below:

X-ma-X	کشمکش	(Kash ma-Kash; Struggle)
	خواہ مخواہ	(Kha ma-Khuwa; Unnecessarily)

X-ba-X	ضرور باضرور	(Zaroor ba-Zaroor; Always)
	خود بخود	(Khud ba-Khud; Itself)
X-a-X	دھڑادھڑ	(Dharr-a-Dharr; Quickly)
	لبالب	(Labalab; Full)

Figure 11: Echo Word Reduplication with ‘م’,‘ب’ or ‘ا’

In some cases the reduplicated word ‘Y’ is either a non-sense word that rhymes with first word ‘X’ or is orthographically or morphologically similar to ‘X’. Y in some cases is formed by changing the vowel of X to /a/. Examples are:

ڈھیلا ڈھالا	(Dheela Dhala; Changing from /ee/ to /a/
ٹھیک ٹھاک	(Theek Thak; -do-)
دھوم دھام	(Dhoom Dham; (Changing from /u/ to /a/)

Figure 12: Echo Word Reduplication Changing Vowels

Y can also be formed replacing /wa/ to X-First Consonant. Examples of such constructions are:

روٹی ووٹی	(Roti Woti)
چابی وابی	(Chabi Wabi)

Figure 13: Echo Word Reduplication Replacing First Consonant

Other consonants instead of /w/ can also be used. Consider following example where /p/ is used instead: ‘اونے پونے’ (Onay Ponay; Cheeply). Urdu grammarians have named Y as ‘محمل’ (Muhmil; non-word). These words do not have meanings of their own but are only used for emphasis [6].

Lastly Y may be a word, starting with the same letter as that of X and related to X in some way. Examples are:

دور دراز	(Dur Daraz; From far away)
پاس پڑوس	(Paas Paros; Neighbourhood)
دن دیہاڑے	(Din Deharay; Durring daylight)

Figure 14: Echo Word Reduplication Rhyming Word

2.5. Compound Verbs or Verb Phrases

In Urdu root verb + intensifying verb combine together to form compound verbs. The root verb or the main verb contains the semantic value of a compound. The intensifying verb (also known compound auxiliary or explicator verb [3]) adds nuance to the meaning of the sequence. Sometimes the meaning of compound verb can not be extracted from its constituent. The compounds in these cases have become lexicalized. Few examples are given below.

مارنا	To Beat
مارڈالنا	To Kill
لینا	To Take
لے جانا	To Take Away

Figure 15: Examples of Compound Verbs

According to [2] compound verbs can be divided into five categories. These are given below with examples.

Table 2: Types of Compound Verbs

Category	Description	Examples
Intensives	The intensifying verb may be transitive or intransitive.	اس نے بچھو کو **مار ڈالا** ‘He **killed** the scorpion’
Potentials and Completive	Must always be constructed actively in tenses composed of the perfect participle	ہم **سن چکے** ہیں ‘We have already heard’
Continuatives (with Imperfect participle)	Formed with inflected imperfect participle and one of the verbs رہنا or جانا	وہ اسی طرح **بکتی رہتی** ہے ‘She keeps on prating in this same way’
Frequentatives or Continuative (with perfect participle) and Desiderative	Are always constructed in the tense composed of the perfect participle	وہ رات بھر پانی میں ہاتھ **مارا کیا** He kept striking his hands in water all night
Transitive	These are formed by conjunctive participle and cannot be passively constructed	چیزوں کو کون **لے گیا** ‘Who took away those things?’

2.6. Urdu Orthography

In order to better understand the Urdu word segmentation problem it is worth spending few paragraphs on the properties of Arabic script i.e. the script in which Urdu is written. Arabic script is written in Right to Left (RTL) direction. Urdu characters[3] change their shapes depending upon neighboring context. But generally they acquire one of these four shapes, namely isolated, initial, medial and final. Urdu characters can be divided into two groups, separators and non-separators. These are also known and non-joiners and joiners respectively. The separators or non-joiners can acquire only isolated and final shape. On contrary non-separators or joiners can acquire all the four shapes. The isolated form of each of these is shown in figures given below.

ا د ڈ ذ ر ڑ ز ژ و ے

Figure 16: Separators/Non-Joiners in Urdu

ب پ ت ٹ ث ج چ ح خ س ش ص ض ط ظ ع غ ف ق ک گ ل م ن ہ ی ھ

Figure 17: Non-Separators/ Joiners in Urdu

Here are the set of rules that the characters use to acquire shapes.

A joining character takes:

[3] Not including diacritics, honorifics and punctuation marks

- Initial form when a joiner follows it (ب -> ﺑ when ج is following like in بج)
- Initial form when a non-joiner follows it (ب -> ﺑ when ر is following like in بر)
- Final form when comes after a joiner (ب ->ﺐ when comes after ج like in جب)
- Isolated form when comes after a non-joiner (ب ->ب when comes after د like in. دب)
- Medial form when it is already in final form and is followed by a joiner (ب -> ﺒ when it is already in final firm ﺐ and is followed by ل like in جب -> جبل)
- Medial form when it is already in final form and is followed by a non-joiner (ب -> ﺒ when it is already in final firm ﺐ and is followed by ر like in جب -> جبر)

A non-joining character takes:

- Isolated form when a joiner or non joiner follows it (د -> د when ج is following دج)
- Isolated form when is followed by a non-joiner (د -> د when it is followed by ر like in در)
- Final form when is followed by a joiner (د ->ﺪ when it is followed by ب like in بد)

Following rules can be algorithmically defined as:

```
For each character in the Input Sequence

    If this character is separator or non-separator

        If previous character is separator or non-separator

            If previous character is non-separator

                Form of this character=final

                If form of previous character is isolated
                    Form of previous character=initial
                Else
                    Form of previous character=medial
                End if

            Else
                Form of this character=isolated
            End if

        Else
            Form of this character= isolated
        End if
    End if
```

Figure 18: Pseudo-code that generally captures Arabic Based Languages

The Urdu text ‘بادشاہی مسجد’ is generated by typing sequence ‘ب ا د ش ا ہ ی م س ج د’. The following table shows how word ‘بادشاہی’ has been formed out of sequence ‘ب ا د ش ا ہ ی’.

Table 3: Step-wise Formation using Algorithm in Figure 3

Input	Output	Description
ب	ب	Isolated form of ب
ب ا	با	ب takes initial form l takes final form
ب ا د	باد	د takes isolated form since **l** is a non-joiner
ب ادش	بادش	ش takes isolated form since د is a non-joiner
ب ا د ش ا	بادشا	ش takes initial form since it's a joiner and also makes **l** to take its final form
ب ا د ش ا ه	بادشاه	ه takes isolated form since l is a non-joiner
ب ا د ش ا ه ی	بادشاهی	Finally ی joins with ه taking final form and making it take initial form

2.6.1. Diacritics

In Urdu diacritics act as dependent vowels and are used along consonants and independent vowels to prolong or stress their sounds. The diacritics (also known as Aerabs) are optionally used in writings. Native speakers can figure out the correct pronunciation of a word by looking at its context or by virtue of their knowledge about Urdu.

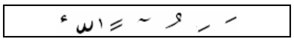


Figure 19: Urdu Diacritics

Aerabs usually stack above or down the text as is shown in following example:

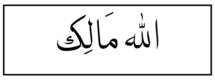


Figure 20: Urdu word with diacritics

2.6.2. Nastalique

Urdu like Arabic is written in Nastalique font which moves upward from the base line in top right direction. See the word formation (left to right) in following example.

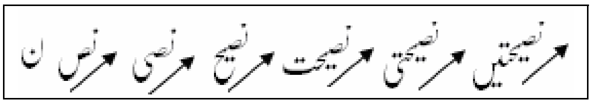


Figure 21: Vertical Movement in Urdu

The text moves upward in top right direction. Because of its cursive nature text written in Nastalique is compact than that written with other fonts.

2.6.3. Concept of Space in Urdu

The notion of space between words is completely alien in Urdu hand-writing. Children are never taught to leave space when starting a new word. They just tacitly use the above listed rules and the human lexicon to know when to join and when to separate. Form example when writing sentence ‘بادشاہی مسجد کا دروازہ بند ہے’ (The door of Badshahi Mosque is closed) a native speaker knows that ‘مسجد’ is a word next to ‘بادشاہی’ so he would start a new word on bases of algorithm given above. Instead of leaving space (like English users do) he would just trigger a rule that ‘مسجد’ is a new word so he does not join ‘م’ (starting letter of ‘مسجد’) with ‘ی’ (final letter of ‘بادشاہی’). If ‘بادشاہی مسجد’ was a single word to user he would write it as ‘بادشاہیمسجد’ . Following is a hand-written sample (written in Nastalique) which shows that space is not used in hand-written Urdu.

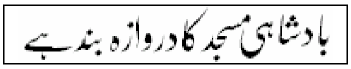


Figure 22: Hand-Written Urdu String

A non-Urdu user can never guess where a word boundary is, just by looking at this text. To him the word boundaries can possibly be like shown by arrows in figure below or he might think it is a single word.

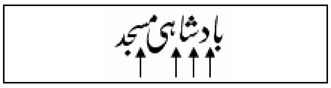


Figure 23: White-Space Word Boundaries

In light of this discussion and the discussion on Urdu morphology and compounding we shall now discuss the word segmentation problems that prevail in Urdu.

3. Word Segmentation Problem in Urdu

As already mentioned that the notion of space character is not common in hand-written Urdu orthography however a machine cannot work like human mind. It must be provided with a separating character to know that ‘بادشاہی’ and ‘مسجد’ are not combined. In other words if the user wants the text to be visually seen as ‘بادشاہی مسجد’ he must provide a computer with a breaking character otherwise it would join and look like ‘بادشاہیمسجد’ which is un-acceptable. Most of the users have accepted the limitation of technology in this case and accepted space as a separating character. In other case where the user does not want the space to be visible uses zero-width non-joiner character (U+200C; ZWNJ). Nevertheless this makes the problem a little relaxed because now the text contains some clues in form of space or ZWNJ about where a potential word boundary is. Space, however does not necessarily means a word boundary, why? We shall see in section ‘Space Deletion’. For now we divide word segmentation into two categories given below i.e. inserting space (or some token at word boundaries) and removing unnecessary spaces.

- Space Insertion
- Space Deletion

3.1. Space Insertion Problem

Space insertion problem arise in case where words are written in continuum without any space or other separating character like ZWNJ. Languages like Chinese, Japanese and Thai solve space insertion problem. Space insertion problem is difficult because there are multiple ways in which a space can be inserted. A classic example from English is famously quoted. There are multiple ways of segmenting following sentence: GODISNOWHERE.

GOD IS NO WHERE	GOD doesn't exist
GOD IS NOWHERE	GOD doesn't exist
GOD IS NOW HERE	GOD is here

Figure 24: Segmentation Ambiguity [5]

In Urdu space insertion problem can be further classified into two sub-categories.

- Non-Joiner Word Ending
- Joiner Word Ending

3.1.1. Non-Joiner Word Ending

We mentioned before that concept of space is uncommon in hand-written Urdu orthography. Then we said that because of the limitation in technology, space (or ZWNJ) has become part of language, and to make the word visually appropriate user must insert something between two words. However, when a word ends with a non-joiner the next word can be written without inserting space. Because non-joiners can not acquire medial and initial shapes; they do not combine with the starting character of next word. This allows the user to start next word without putting space. Consider the same example that has been given above. A native speaker may or may not put space between 'کا' and 'دروازہ' because 'کا' ends with a non-joiner 'ا' and will not connect with 'د' (first character of following word). So without space 'کادروازہ' is as much acceptable as 'کا دروازہ'. Therefore a sentence with all words ending with non-joiners might not have space character at all. One such example is shown below.

(a) قافلےکےلیڈراحمدشیرڈوگرنےکہا	(b) قافلے کے لیڈر احمد شیر ڈوگر نے کہا
Troop of leader Ahmad Sher Dogar said	
Troop leader Ahmad Sher Dogar said	

Figure 25: Sentence with Non-Joiner Word Endings (a) With no Space (b) With Spaces

As can be seen (a) and (b) look visually identical although (a) doesn't have any space while (b) has space after each word. Ambiguity arises when a word is composed of smaller words and is required to be segmented differently based on context it is occurring. Example is shown below.

نوجوان ادھر او	پنجاب برگیڈ کے نوجوان شہید ہوے ہیں	وہ نوجوان کے ساتھ ہر جگہ جاتے ہیں
Young lad come here	Punjab Brigade of nine soldiers martyred have been	Those nine that them with every place go
Young lad come here	Nine soldiers of Punjab brigade have been martyred	Those nine that go with them every place

Figure 26: Three Possible Segmentations of Word 'نوجوان'

In first sentence 'نوجوان' is a single word that means 'young lad' or 'youngster'. In second example 'نوجوان' is composed of two words 'نو' and 'جوان' which mean 'nine' and 'soldiers' respectively. There is an alternative translation of second example. It may also mean 'The soldiers of Punjab brigade have been martyred' in which case 'نوجوان' is again a single word which means 'soldiers'. In last scenario 'نوجوان' is composed of three words 'نو', 'جو' and 'ان' which mean 'nine', 'that' and 'them'. Last word 'ان' is usually are pronounced as 'اُن' (un; them) or 'اِن' (in; them). Pronunciation in Urdu are marked by diacritics which in above case is Pesh ' ُ ' (stacks above character) or Zer ' ِ ' (connects to the bottom of a character). However, the use of diacritics has become rare in Urdu text because a native speaker can guess the pronunciation through tacit knowledge or by looking at the context of word. Third scenario would have been ruled out if the text was diacritized.

Non-joiner word ending, space insertion problem is initiated by Urdu orthography. It can occur in all kinds of words that end with non-joiners. Not putting space has almost same visual impact. This makes its use optional and the user might only put it for the sake of tidiness/readability.

3.1.2. Joiner Word Ending

As mentioned afore users put space before writing next word if previous word ends with a joining character. This practice is largely followed but not always. In some cases native speakers may prefer joining them. This is perhaps because they perceive the two as a single word. As a result of this both joined and separated version exist in corpus. One may chose one way or another depending upon their conception. Below are given examples of such word.

3.1.2.1. Oblique Pronouns

Oblique pronouns when followed by postposition ‘کو’ (to), ‘کا’ (of), ‘سے’ (from) are perceived to be a single unit. Considered following constructions:

مجھ کو	مجھکو	مجھے
تجھ کو	تجھکو	تجھے
جس کو	جسکو	جسے

Figure 27: Oblique Pronouns with ‘کو’ Construction

Another reason to consider these as a single word is that ‘مجھ کو’ or ‘مجھکو’ is used alternatively to ‘مجھے’ which is a single word. Usage of former is obsolete, later is more common these days [3]. In fact ‘مجھ کو’ construction is considered invalid by some linguists. Primary class students are asked to correct ‘مجھ کو’ construction to ‘مجھے’ in grammar exams. According to [2] ‘مجھکو’ (to me) is used in dative construction whereas ‘مجھے’ (me) is accusative and first person pronouns. The analysis of Platts [2] is influenced by Hindi analysis where case markers are considered to be part of a word. However Schmidt [3] classifies these as oblique pronoun + ‘کو’ construction. Another reason for considering it two words is that a proper noun can be used in place of ‘مجھ’. Consider following construction:

a	مجھے آم بہت پسند ہیں	I like mangoes
b	مجھ کو آم بہت پسند ہیں	I like mangoes
c	اسلم کو آم بہت پسند ہیں	Aslam like mangoes
d	* اسلم آم بہت پسند ہیں	Invalid Construction

Figure 28: Replacing Proper Noun with Pronoun

If ‘کو’ was part of ‘مجھ’ then it would have been completely replaced by proper noun ‘اسلم’ as in case of (d) which is an invalid construction. In light of these arguments it is more rational to consider these as two words. A comprehensive list is shown in Appendix A.

3.1.2.2. Possessive Pronouns

Similar situation arise in case of possessive forms of personal pronouns. The first person and second person pronouns ‘میرا’ (mine) and ‘تمھارا’ (yours) respectively are single word. Based on that it is reasonable to consider second person (respect level 1) ‘آپ کا’/‘آپکا’ (yours) and third person pronouns ‘ان کا’/‘انکا’ (theirs) as a single word. On contrary replacement with proper noun test provides a counter argument. Consider following sentence construction.

a	تمھارا کیا خیال ہے	What is your opinion
b	آپ کا کیا خیال ہے	What is your opinion
c	اسلم کا کیا خیال ہے	What is Aslam’s opinion

Figure 29: Replacing Proper Noun with Pronoun

Again the proper noun ‘اسلم’ has only replaced ‘آپ’ and not ‘آپ کا’. Again [1] & [2] has classified these as one word; these are mentioned as possessive personal pronouns in [3]. We categorize these as two separate words. Complete list is given in Appendix A.

3.1.2.3. Adverbs or Adverbial Phrases

When oblique nouns are preceded by oblique singular demonstrative ‘اس’ or oblique of ‘کیا’ (kya) or ‘جو’ (jo) they form adverbs or adverbial phrases. As a result these are written in joined and separated forms. These are classified as adverbs in [3]. [1] & [2] have not mentioned about such cases. Below are such cases.

a	اسوقت صرف وہی میرے کام ایا	He was the only one who helped me at that time
b	اس وقت صرف وہی میرے کام ایا	He was the only one who helped me at that time
c	کعبہ کسطرف ہے	What is the direction of kaba[4]
d	کعبہ کس طرف ہے	What is the direction of kaba
e	اس طرح کرو	Do it this way
f	اسطرح کرو	Do it this way

Figure30: Adverbs: Time (a) & (b), Place (c) & (d), Manner (e) & (f)

In above figure (b) provides ‘with space’ version of ‘اس وقت’ (that time) while (a) gives its ‘without space’ form ‘اسوقت’ both forms exist in corpus. We consider these to be composed of two words. Consider following construction.

اس برے وقت صرف وہی میرے کام ایا	He was the only one who helped me at that hard time

Figure 31: Adjective Inserted between Demonstrative ‘اس’ and Noun ‘وقت’

An adjective ‘برے’ (hard/difficult/bad) is inserted between demonstrative ‘اس’ (that) and noun ‘وقت’ (time). ‘اس برے وقت’ (at that difficult time) is an adverbial phase and so is ‘اس وقت’ (at that time). Similarly ‘کسطرف’ and ‘اسطرح’ each is composed of two words ‘کس+طرف’ and ‘اس+طرح’ respectively. A list for few such adverbial phrases is given in Appendix B.

In another case ‘یہاں پر’ (Over here) is alternatively written as ‘یہانپر’. This creates a complex scenario because ‘ں’ (Arabic Letter NOON GHUNAH; U+06BA) is changed to ‘ن’ (Arabic Letter NOON; U+0646) when written in joined form. This problem is very hard to address.

3.1.2.4. Compound Postpositions or Postpositional Phrases

Some postpositions that are originally feminine nouns demand ‘کی/کے’ (ki / ke) with the genitive they govern. There is confusion whether ‘کی’/’کے’ is part of following postposition or not. As a

[4] Khana Kaba the place where Muslims perform Hajj. Prayer is offered in the direction of Kaba.

consequence, word ‘کی طرف’ (towards) is also written as ‘کیطرف’. According to [2] ‘کی’ is an affix to postposition ‘طرف’. According to [3] these can be classified as compound postpositions or postpositional phrases. Apparently it seems to be a single word but it is actually used in KA + Oblique NOUN construction. The case marker ‘کا’ (ka) is inflected to ‘کی‘(ki) when making agreement with noun in gender and case.

A more confusing situation can arise where a sentence has such construction accompanied with possessive pronouns ‘اپ’ (discussed above). Consider following phrasal construction‘اپ کی طرف’ (towards you) and its written variations.

a	اپکی طرف	‘کی’ is connected to possessive pronoun ‘اپ’
b	اپ کیطرف	‘کی’ is connected to oblique noun ‘طرف’
c	اپ کی طرف	Plain construction with no joining

Figure 32: Postpositional Phrases (Varied Constructions)

Figure (a) considers ‘اپکی’ as a single pronoun, (b) considers ‘کیطرف’ as a single affix ‘کی’ attached with postposition ‘طرف’. This shows function words are combined with content words to get meaningful piece. These are separate words nevertheless. We consider the last one most appropriate. Appendix C gives lists some of the problem children in this category.

A more interesting variation of this problem occurs in some cases which are spelled differently when written in joined form. For example ‘کے خلاف’ (against) is alternatively written as ‘کیخلاف’ where ‘ے’ (Arabic Letter YEH BAREE; U+06D2) is converted to ‘ی’ (Arabic Letter Farsi YEH; U+06CC) when written jointly.

3.1.2.5. Compound Verbs or Verb Phrases

Helping verbs (also known as vector or intensifying verb) when jell with the root verb loose their lexical meaning to some extent, but adds a nuance to the root verb. The function of helping verb is to show tense and agreement. In some cases helping verbs (modals and auxiliaries) are written with root verbs without space. For example ‘کرے گی’ (will do) is alternatively written as ‘کریگی’, ‘دے دیا’ (given) is alternatively written as ‘دیدیا’. Even in these cases both have different spellings. ‘ے’ (Arabic Letter YEH BAREE; U+06D2) is converted to ‘ی’ (Arabic Letter Farsi YEH; U+06CC) when written jointly. [2] & [3] have classified these as compound verbs. [1] has mentioned these as ‘فعل ناقص ’ (Empty Verbs) and ‘افعال معاون’ helping verbs. We will agree with [1]. Separating these into two words is more helpful in syntactic and semantic analysis. Appendix D lists some of such problem verb phrases.

3.2. Space Deletion Problem

Space deletion problem is second part of Urdu word segmentation. Space insertion is a widely studied predicament because most of the Asian languages lack space. Space deletion, however is uncommon in other Asian languages. In Urdu, as already mention there is no concept of space. Space is only inserted in printed texts to avoid joining between two words because there is no other way out. However, space is not only used as a words separator. There are a lot of cases in which a single word is orthographically written without joining. In order to achieve that visual impact computer users put space (or ZWNJ) in between. The space insertion problem commonly occurs in following types of words.

- Words with derivational affixes
- Compound Words
- Proper Nouns
- English Words

- English Abbreviations

3.2.1. Derivational Affixation

Derivational Affixes are common part of Urdu. Both prefixation and suffixation cause space deletion problem. If the first part (stem or affix) of word ends with a non-joiner native speaker might or might not put space because it appears visually identical. Example is 'غیرضروری' (unnecessary). The first part of this word 'غیر' ends with a non-joiner 'ر' so the user might or not put space before writing 'ضروری'**.** On contrary if first part of word ends with a joiner then the user must use space to separate them for the sake of readability perhaps. For example in word ' حیرت انگیز' (amazing) 'حیرت' ends with a joiner 'ت' so the user will always insert a space before 'انگیز'. So 'حیرت انگیز' is never written as 'حیرتانگیز' because it seems unreadable. Some of the examples are given below while few others cane be seen in Appendix E.

Prefixation		Suffixation	
بےچینی خوش نصیب	Anxiety Fortunate	منصوبہ بندی سرمایہ کاری	Planning Investment

Figure 33: Space Insertion Problem Derivational Affixation

However, there are a few exceptions in which first part (stem or affix) ends with a joiner but user still doesn't put space. To make it more complex the joined form and separated forms have different spellings. For example 'مزے دار' (delicious) is alternatively written as 'مزیدار' where 'ے' is converted to 'ی'.

Most of the cases have at most one derivational prefix or derivational suffix but there are a few exceptions where a word has more than one derivational affixes. For example 'غیرشادی شدہ' (unmarried) consists of stem 'شادی' (marriage), 'غیر' (un) and 'شدہ' (gives a sense that he/she is unmarried or married in case of plain 'شادی شدہ'.

There are some cases in which affixes themselves may exist as free morphemes. In one sentence it can occur as a prefix or suffix to a word as in shown in examples below where 'خوش' and 'ناک' occurs as prefix and suffix respectively.

(a) وہ بہت خطرناک ہے	(b) ہر کوی اپ سا خوش نصیب نہیں
He very dangerous is	Every one you like fortunate not
He is a very dangerous	Not every one is fortunate like you

Figure 34: (a) Suffix 'ناک' (b) Prefix 'خوش'

On contrary 'ناک' and 'خوش' can occur as free morphemes in which case they mean 'nose' and 'happy'. Given are examples.

اس کی ناک سے خون بہ رہا تھا	وہ اس دن واقع ہی خوش تھا
His of nose blood flowing was	He that day really happy was
His nose was bleeding	He was really happy that day

Figure 35: Free Morphemes 'ناک' and 'خوش'

3.2.1.1. Wala (والا) Suffix or Wala Phrase

Wala is categorized as suffix in most grammar books. It is commonly employed to form noun of agency (اسم فاعل), possession and various other relations [6]. In different constructions it may attach to oblique infinitive, oblique noun or an adjective or adverb [3]. These constructions are given below.

3.2.1.1.1. Oblique infinitive + Wala

In this case wala agrees with the following noun. Example is given below.

گیت گانے والی لڑکی کون ہے
Song singing girl who is
Who is the girl singing the song

Figure 36: Oblique Infinitive + Wala [3]

Although Wala forms a noun combining with oblique infinitive 'گانے' to form the noun doer (singer in this case) but Wala can not be considered to be part of 'گانے'. It is rather covering the entire phrase 'گیت گانا'. Let us see this in light of syntax trees.

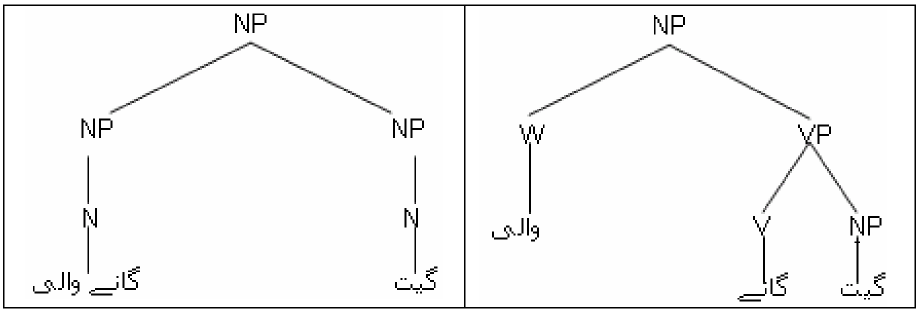


Figure 37: (a) Wala Part of 'گانے' (b) Wala Connects to VP

(b) gives the proper syntax tree for the phrase 'گیت گانے والی'.

3.2.1.1.2. Oblique Noun + Wala

Wala following an oblique noun makes an adjective phrase [3]. Consider following phrase:

میری سبز رنگ والی کتاب کہاں ہے ؟
My green color book where is?
Where is my book with the green color?

Figure 38: Oblique Noun+ Wala

Also in this case 'والی' connects to adjective phrase 'سبز رنگ' (green color) and form another adjective phrase.

3.2.1.1.3. Occupational Nouns

Wala forms occupational nouns [2]. Examples are given below.

پولیس والا	Policeman
دودھ والا	Milkman
رکشے والا	Rikshaw Driver
گانے والی	Singer

Figure 39: Oblique Noun+ Wala [3]

In this case 'والا' is treated as a single word. Note that 'گانے والی' (singer) is a single word in this case because it is signifying the profession.

3.2.1.1.4. Adjective/Adverb + Wala

Wala follows adjective or adverb to form their respective phrases. This construction is only used in spoken and not in formal texts. Wala is treated as separate word in this case swell.

اس نے رات والی کہانی دوہرا دی	اس نے میرا مہنگا والا لباس خراب کردیا
He night of story repeated	He mine expensive one dress ruined
He repeated the story he told at night	He ruined my expensive dress

Figure 40: (a) Adverb + Wala (b) Adjective + Wala

So Wala is a separate entity in all cases except where it forms occupational nouns.

3.2.2. Compound Words

All categories of compound words have been already discussed in section 'Compound words in Urdu'. However all of these can not be treated as a single unit. This section revisits each of these.

3.2.2.1. XY Formation

We will treat the words in XY compounds as single word because the two morphemes combine together to form a new semantic entity. Examples are given below. If the first morpheme ends with a non-joiner then user may or may not put space depending upon his conception whether he consider it as a single unit or two words or for the sake of readability. Examples are given below.

پڑھالکھا vs. پڑھا لکھا	Educated
روزنامہ vs. روز نامہ	Newspaper
گھوڑڈوڑ vs. گھوڑ ڈوڑ	Horse Race

Figure 41: Non-Joiner First Morpheme Compound

Space must be inserted when first morpheme ends with a joiner as shown below.

ماں باپ	Parents
ناک نقشہ	Features
دیس نکالا	Exiled

Figure 42: Joiner First Morpheme Compound

However, there are still a few cases in which a word ends with a joiner but user may or may separate depending upon personal preference. As a result both versions of such words exist. Examples are shown below.

ڈاکخانہ vs. ڈاک خانہ	Post Office
جبکہ vs. جب کہ	Although

Figure 43: Joiner First Morpheme Compound Joined Vs. Separated

To further complicate the matters there are few cases in which a compound is spelled differently when written in joined or split apart form. Given are examples below.

کیوں کر vs. کیونکر	How
کیوں کہ vs. کیونکہ	Because

Figure 44: Spelling Variation in Joined and Non-Joined Form

In both these scenarios ‘ں’ (Arabic Letter NOON GHUNAH; U+06BA) is changed to ‘ن’ (Arabic Letter NOON; U+0646). This problem is very hard to address. It can not be classified as a spell checking problem because both variations are popularly used and sometimes users prefer one over another for a reason say using ‘کیونکہ’ to save space or because it is a single word.

3.2.2.2. X-o-Y Formation

The linking morpheme is –o– is very productively used to form compounds in Urdu. The words on both sides are usually Arabic or Farsi. Nowadays it is frequently used even to join English words and the linking morpheme itself is used in the sense of ‘and’. It is hard to decide whether these two are to be treated as a single or as multiple words. Consider the following examples.

Table 4: Compound Words with Linking Morpheme –و–

Column-I	Column-I	Column-II	Column-II
قومی و صوبائی	National and Provincial	نشونما	Upbringing
علما و مشائخ	Scholars and Philosophers	نظم و ضبط	Discipline
دینی و سیاسی	Religious and Political	عزیز و اقارب	Relatives
ڈسٹرکٹ و سیشن	District and Session	تباہ و برباد	Destroy
پٹرول و ڈیزل	Petrol and Diesel	امن و امان	Peace
حج و عمرہ	Hajj and Umrah	امد و رفت	Traffic
اسلم و عمران	Aslam and Imran	تعلیم و تربیت	Education

Words in column-I are clearly two words connected by linking morpheme –و– which means ‘اور’ (and). All the examples given in above figure contain two words but –و– can be used very effectively to join any number of words. Example is ‘دینی و مذہبی و سیاسی و سماجی کارکن’ (Religious, political and social worker) .The words in column-II are closely jelled, although –و– in these also give sense of ‘and’. To most Urdu users these are lexicalized nevertheless. Therefore we assume these as single units.

3.2.2.3. X-e-Y Formation

As already mentioned the linking morpheme –e– signifies the possessor relation in which X belongs to Y. In alternative construction case markers ‘کا/کی’ (ke/ka) are used to mark possession. For example ‘حکومت پاکستان’ (Government of Pakistan) can be alternatively written as ‘پاکستان کی حکومت’. Similarly ‘مرکز تحقیقات اردو’ (Center for Research in Urdu) can be written as ‘ارد تحقیقات کا مرکز’. Since these are used very productively we treat these as separate words.

Izafat is also used for noun-e-adjective formation in which X is modified by adjective Y. Examples are ‘مہمان خصوصی’ (chief guest) and ‘وزیر اعظم’ (prime minister). These are lexicalized entries in the mind of a native speaker. In figure below words in column-I are treated as two words and the ones in column-II are treated as single word.

Table 5: Compound Words with Linking Morpheme –e–

Column-I	Column-I	Column-II	Column-II
حکومت سندھ	Government of Sindh	وزیر اعلیٰ	Provisional minister
وزیر صحت	Minister of Health	وزیر اعظم	Prime minister
قائد حزب اختلاف	Opposition Leader	نشاط ثانیہ	Renaissance
نماز عشاہ	Isha Prayer	دیوان خاص	Drawing room
زیر غور	Under consideration	سنگ میل	Milestone
نظام تعلیم	Education System	حزب اختلاف	Opposition
تعلیم نسواں	Women Education	جلسہ عام	Procession

The linking morphemes –o– and –e– are very productively used to form compound words. Some compounds involve both of these. Few such examples are ‘فضائل و مسائل حج و عمرہ’ (Blessings and Problems in Hajj and Umrah) and ‘اسمائے مکان و زماں’ (Nouns of place and time).

3.2.2.4. Reduplication

All forms of reduplication discussed above are treated as single word because reduplication is a morphological process.

3.2.3. Proper Nouns

Often the names of places or personalities are written with space in between. For example, country name ‘سعودی عرب’ (Saudi Arabia) is written with space between ‘سعودی’ (Saudi) and ‘عرب’ (Arab). This again creates ambiguity because ‘سعودی’ and ‘عرب’ can also exist as separate morphemes in which case ‘سعودی’ is referring to some one who has nationality of ‘Saudi Arabia’ and ‘عرب’ can independently refer to Arab but together they mean country name. Another similar example is ‘وزیرآباد’ (Wazirabad) which name of a city of Pakistan. Also in this case both the morphemes ‘وزیر’ (Vizier) and ‘آباد’ (Developed) can exist independently.

Some names are also written with space in between. ‘انعام اللہ’ (Inamullah) is one such example. We have mentioned it as ‘Inamullah’ in English the /u/ sound in ‘ullah’ is because of diacritic PESH ‘ ُ ‘ on Alef ‘ا’. But as mentioned before diacritics are not so common in texts so a non-Urdu speaker can read it as ‘ullah’, ‘illah’ or ‘Allah’. ‘illah’ is a non-word but ‘Allah’ is a valid word. This creates ambiguity whether it is part of name or ‘Allah’ (GOD) because ‘انعام’ alone is a valid name. Consider an example below where a single sentence can have two different meanings.

انعام اللہ کے سوا کسی سے نہیں ڈرتا	انعام اللہ کے سوا کسی سے نہیں ڈرتا
Inam Allah than other anyone not afraid	(I/You/He) Inamullah other than anyone not afraid
Inam is afraid of no one but Allah	I/You/He am/are/is afraid of no one but Inamullah

Figure 45: Ambiguity in Proper Names

Second construction is less common but is still found. In this sentence the speaker is talking about someone who is only afraid of Inamullah. But from sentence it is not clear who is being talked about. It can be first person (himself), second person (the listener of the sentence) or third person (someone else).

3.2.4. English Words

Some words adapted from English are now very commonly used in Urdu. In some cases space must be inserted where other are written both in joined and separated form. These are shown in following table.

Table 6: English Words Transliterated in Urdu (Joiners)

Joiners (Always Separated)	
نیٹ ورک	Network
شپ چیمپئن	Championship
Joiners (Separated or Joined)	
ٹیلی فون vs. ٹیلیفون	Telephone
یونی ورسٹی vs. یونیورسٹی	University
فٹبال vs. فٹ بال	Football
Non-Joiners (Separated or Joined)	
موٹر سائیکل vs. موٹرسائیکل	Motorcycle
واٹر بورڈ vs. واٹربورڈ	Water board

3.2.5. English Abbreviations

Abbreviations are not used in Urdu but English abbreviations are used. The pronunciation of each English letter in abbreviation is written in Urdu and each letter is separated by space. In essence an abbreviation is a single word. So this is a space deletion problem. Examples are given below.

پی ایچ ڈی	PhD
پی ائ اے	PIA
ایم کیو ایم	MQM

Figure 46: Abbreviations

Abbreviations are not necessarily a single word. In case of names of person each letter of abbreviation is a separate word because it represents a constituent of name (initial, medial or family) and each part of a name is a separate word according to our analysis. Examples are given below.

ایس ایم ظفر	S.M.Zafar
این ڈی شاکر	N.D.Shakir
ایس اے قریشی	S.A.Qureshi

Figure 47: Abbreviations in Person Names

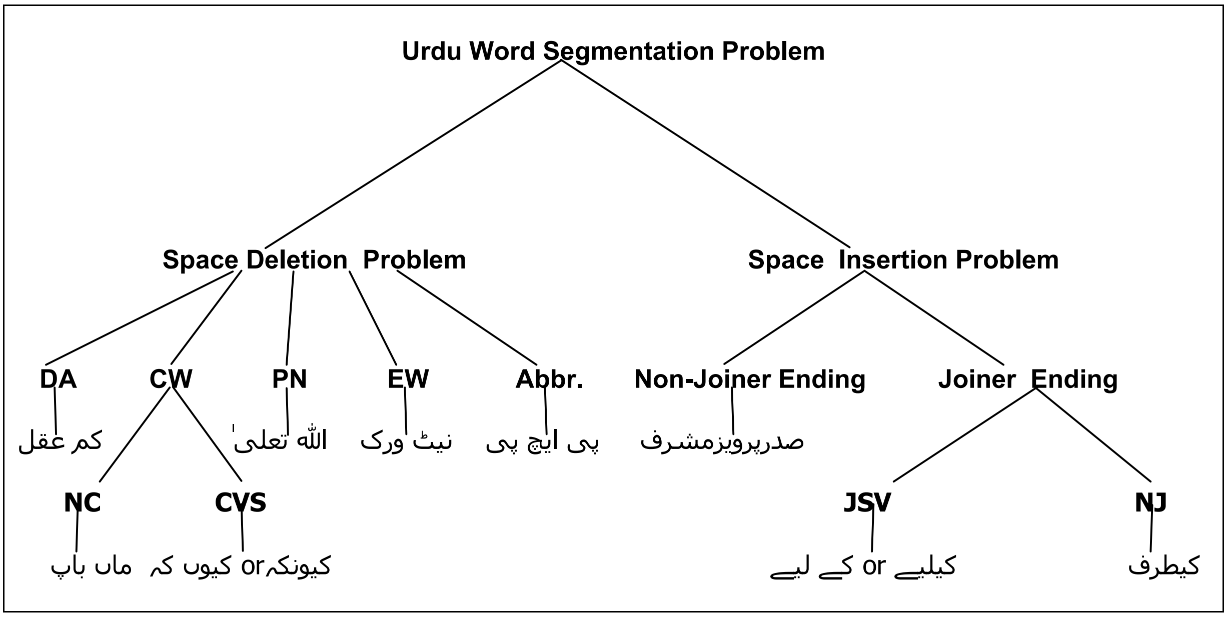


Figure 48: Urdu Word Segmentation in Nutshell

Key: DA=Derivational Affixes, CW=Compound Words, PN=Proper Nouns,
EW=English Words, Abbr. =Abbreviation, NC=Normal Compounds
CVS=Compounds with Spelling Variation, JSV= Joiners with Spelling Variation,
NJ=Normal Joiners

4. Word Hierarchy

In this section we now concretely define word. As already seen defining word is very tricky and no consensus can be reached unanimously. So we categorize words at different levels. If we carefully examine, a sentence can be broken as shown in following figure:

Letters	‘T’ ‘r’’ y’ ‘t’ ‘o’ ‘u’ ‘n’ ‘d’ ‘o’	'م' 'ی' 'ں' 'ض' 'ر' 'و' 'ر' 'ت' 'م' 'ن' 'د' 'ہ' 'و' 'ں'
↓	↓	↓
Morphemes	‘Try’ ‘to’ ‘un’ ‘do’	'میں' 'ضرورت' 'مند' 'ہوں'
↓	↓	↓
Words	‘Try’ ‘to’ ‘undo’	'میں' 'ضرورت مند' 'ہوں'
↓	↓	↓
Phrases	‘Try’ ‘to undo’	'میں' 'ضرورت مند ہوں'
↓	↓	↓
Sentence	‘Try to undo’	'میں ضرورت مند ہوں'

Figure 49: Letter to Sentence

4.1. Compounding

Letters make morphemes, morphemes make words, words make phrases and phrases make sentences. But life is not as simple as it seems. There are some categories that lie somewhere in between phrases and words. We don’t know whether to classify these as words or as phrases because some of the instances of this class seem to be word themselves while others are clearly phrases without functional words. Former are so tightly jelled that they are lexicalized into the minds of native speakers whereas later are not and appear to be phrases. For example a native speaker would always say that ‘وزیر اعظم’ (Prime minister), ‘موم بتی’ (Candle) and ‘نظم وضبط’ (Discipline) as a single entity and is not so sure about ‘مرکز تحقیقات اردو’ (Center for Research in Urdu), ‘ماں باپ’ (Parents) and ‘دینی و سیاسی’ (Religious and Political). First category of examples appears to be word like, where as second class seems to fall in the category of phrases, where ‘مرکز تحقیقات اردو’ is same as ‘اردو تحقیق کا مرکز’, ‘ماں باپ’ is same as ‘ماں اور باپ’ , ‘دینی و سیاسی’ is same as ‘دینی اور سیاسی’. Similar construction can not be devised for first category of examples. But both the categories fall under heading of compounds.

So we add an intermediate level named compound in figure 49 without classifying these as words or phrases. We will also treat words defined in section 3.23 and 3.24 under this category. Therefore words like ‘ٹیلی فون’ and ‘انعام اللّٰہ’ fall under this category.

4.2. Reduplication

Urdu richly exhibits reduplication. Again there is confusion because some examples are purely deemed to be as words where as others are classified as a product of a word ‘X’ plus variation of word ‘X’ which itself is a non-word that is orthographically or phonetically similar to ‘X’. A native speaker has no doubt in his/her mind that ‘کشمکش’ and ‘برابر’ where as they would not be very sure about ‘لٹھم لٹھا’ and ‘پانی وانی’. Former category of examples is lexicalized into minds of native speaker whereas later is not. Therefore again we define an intermediate level between word and phrases known as reduplication without committing anything about their word hood.

4.3. Abbreviations

Like compounding and reduplication, abbreviations are also a source of confusion where we do not know whether to classify these as words. There is no general consensus on whether ‘ پی ائی اے’ is a one word or three. So we define an intermediate level between words and phrases that deals with abbreviations. Abbreviations used in names are treated as separate words because they signify initial and medial names. Therefore ‘پی ائی اے’ is identified as one unit where as ‘این ڈی شاکر’ is classified as three.

Having solved these three problem areas we now redefine our picture that we will use to model our problem.

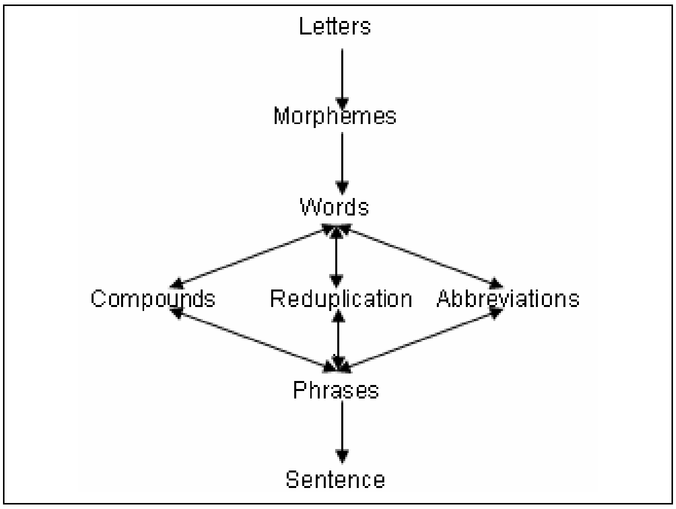


Figure 50: Letter to Sentence-Modified

If we further streamline the above given figure we would be able to draw a clearer picture based on which a model can be defined.

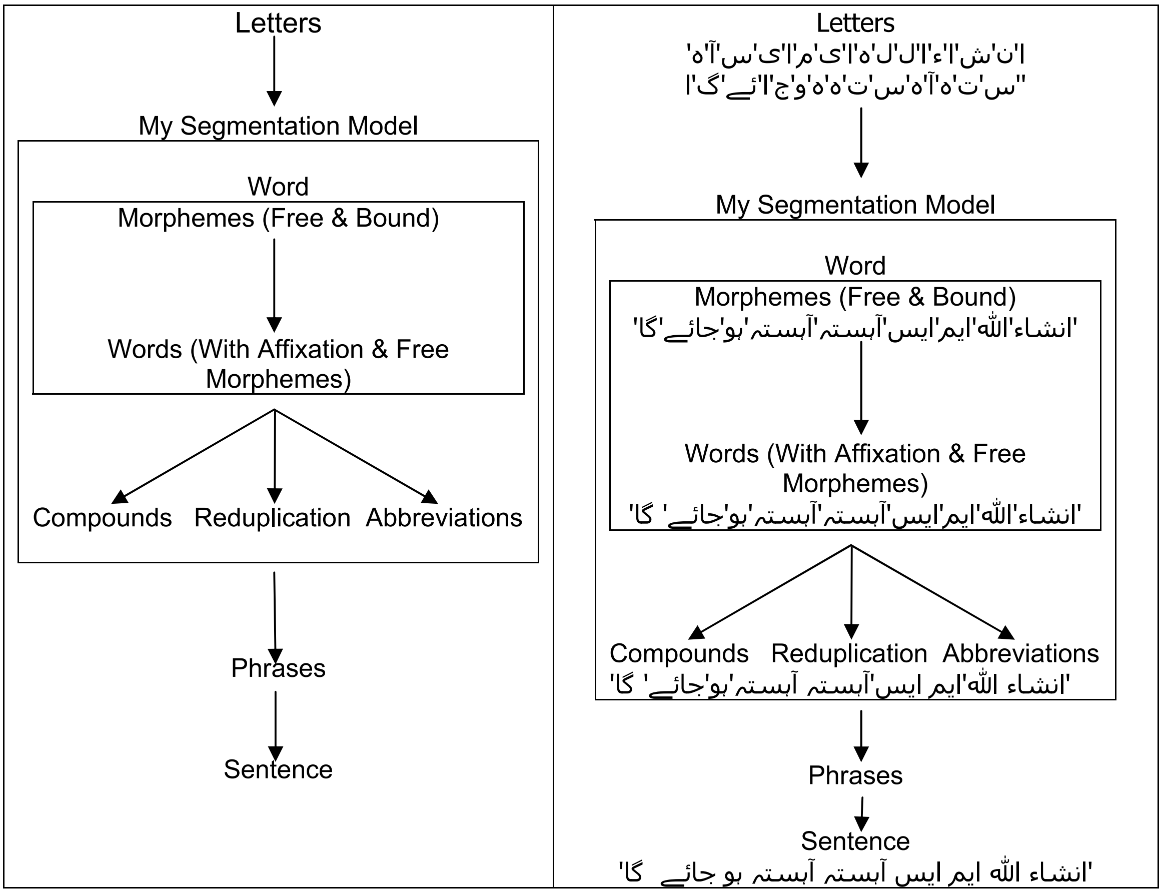

Letters
My Segmentation Model
Word
Morphemes (Free & Bound)
Words (With Affixation & Free Morphemes)
Compounds
Reduplication
Abbreviations
Phrases
Sentence
Letters
ا'ن'ش'ا'ء'ا'ل'ل'ہ'ا'ی'م'ا'ی'س'آ'ہ'
"س'ت'ہ'آ'ہ'س'ت'ہ'و'ج'ا'ئے'گ'ا'
My Segmentation Model
Word
Morphemes (Free & Bound)
'انشاء'الله'ایم'ایس'آہستہ'آہستہ'ہو'جائے'گا'
Words (With Affixation & Free Morphemes)
'انشاء'الله'ایم'ایس'آہستہ'آہستہ'ہو'جائے' گا'
Compounds
Reduplication
Abbreviations
'انشاء الله'ایم ایس'آہستہ آہستہ'ہو'جائے' گا'
Phrases
Sentence
'انشاء الله ایم ایس آہستہ آہستہ ہو جائے گا'

Figure 51: Letter to Sentence-Modified-Reloaded

By defining a hierarchy such as above helps define a model that can produce an output at different level. At level 1 the model simple generates morphemes (free and bound both), generates constituents of compounds, abbreviations and reduplications. At second level the model joins free and bound morphemes to generate words with affixation. For example 'ضرورت' and 'مند' will be joined at this level. As shown in figure above, the model clearly says that the output up till this level lies inside the box known as word. This is because there is no dispute over the word hood of these. At a third level of output the model joins the compound words, reduplication and abbreviations.

5. Problem Statement

The statement of problem for the thesis work presented in this document is:

> *"Given a sentence of valid Urdu words that have space insertion and deletion problems, detect word boundaries based on definition of word modeled in section 4, figure 51"*

This work goes beyond word level and also detects boundaries at compound, abbreviation and reduplication level and outputs sentence at three different levels.

6. Literature Review on Existing Techniques

The techniques previously used can be roughly classified into three categories:

- Lexical Rule Based
- Statistical Approach
- Feature Based Approach

This section briefly traverses through various different techniques under these categories. The hybrid approach is a more recent phenomenon that combines lexical knowledge with statistical information.

6.1. Lexical Rule Based Approach

Rule based approach makes use of lexical knowledge to perform segmentation. Most commonly used models are:

- Longest Matching
- Maximal Matching

6.1.1. Longest Matching

Most early works in word segmentation are based on longest matching [16, 17]. Longest matching technique scans the input from left to right (or right to left for Arabic Script) and tries to find the longest possible match with in dictionary. If the match is found at nth the next search begins from (n+1)th character in the input string. In case the algorithm fails to find rest of the words in the sentence the algorithm must back track to find the next possible match.

Linguistic information, combined with longest matching is employed by [18] to speed-up Lao word segmentation. The algorithm first identifies syllable boundaries in text through a finite set of rules.

A word in Lao can have 1-5 syllables; the algorithm tries to combine up to 5 syllables and perform a look-up in dictionary. Identifying the syllables first improves efficiency because it performs lesser number of dictionary look-ups.

Longest matching is used with word binding force in [19] for Chinese word segmentation. Words in lexicon are divided into 5 groups having 1, 2, 3, 4 or more than 4 characters. Because Chinese words mostly have one or two characters. Searching for longer words as practiced in longest matching is an extremely inefficient approach. To solve this problem lexicon is reorganized so as all the entries in lexicon are structured to have one or two letters. The words having 3 or more characters are broken into pieces. The pieces are stored as free morphemes, affixes or infixes. Each entry as a pointer to all its possible affixes infixes. These entries are than coalesced so as to find longest match.

Longest matching fails to find correct segmentation because of its greedy characteristic. A classic example from Thai word segmentation ไปหามเหลี (go to see the queen) is incorrectly segmented as ไป (go), หาม (carry), เห (deviate), ลี (color). However the required segmentation is ไป (go), หา (see), มเหลี (queen) [16].

6.1.2. Maximum Matching

Maximum matching algorithm was proposed to solve the shortcomings of longest matching. Unlike longest matching algorithm it generates all possible segmentations for a given input and select the one that contains fewest words. This can be efficiently achieved through dynamic programming technique. So maximum matching will correctly segment ไปหามเหลี into ไป (go), หา (see), มเหลี (queen) as it contains lesser number of words.

Because the algorithm uses global maximum matching rather than using local greedy heuristics it always outperforms longest matching technique. The algorithm will fail in the case when alternatives have same number of words, as it can not determine the best candidate. Some other heuristics are often applied then. These heuristics might again be greedy one for example preferring the longest matching at each point [20]. Minor variants of maximum matching are discussed in [25, 26, 27 and 28].

Longest and maximum matching approaches prefer compound words over simple words. Maximum matching prefers overall number of words to be minimum. With a fully comprehensive dictionary above 95% accuracy can be obtained by using longest and maximum matching techniques. [22] has reported 98% accurate results with 1300 simple sentences. However, it is obvious that it is impossible to have such a dictionary having all morphological forms of word. Dictionaries normally keep one entry, known as base form, for each word. Even if it contains all of its possible citation forms we can't expect it to contain all possible personal pronouns and transliterated foreign words. The efficiency of these techniques drops adversely when the input text contains unknown words. This is shown in figure below. The data shown is based on Thai word segmentation problem.

Table 7: The Accuracy of Two Dictionary-Based Techniques vs. %age of Unknown Words [21]

Unknown Word (%)	Accuracy (%)	
	Maximal Matching	Longest Matching
0	97.24	97.03
5	95.92	95.63
10	93.12	92.23
15	89.99	87.97
20	86.21	82.60

25	78.40	74.41
30	68.07	64.52
35	69.23	62.21
40	61.53	57.21
45	57.33	54.84
50	54.01	48.67

It is evident that the accuracy drops drastically as the percentage of unknown words increases. With 50% unknown words, accuracy for both maximal and longest matching techniques declines to 54 and 48% respectively.

The second problem with these two techniques is their dealing with segmentation ambiguities. Longest matching algorithm deals with ambiguity simply by ignoring it. The method is guaranteed to produce just a single segmentation. Maximum matching on contrary has to provide criteria for choosing best out of a set of multiple possible segmentations. Some of these criteria might be based on syntactic or semantic features (e.g. [23] that use a unification approach). Others are based on different lexical heuristics. For example [24] attempts to balance the length of words in a three word window, preferring segmentation that give approximately equal length for each word. Nevertheless, no single criterion or a set of criteria can cater all the possible segmentation ambiguities and therefore some of these might still be incorrectly resolved. Consider following example

Input	Possible Segmentations	
نوجوان	نوجوان	Young Lad
	(youngsters)جوان (nine) نو	Nine youngsters
	(them)ان (that) جو (nine) نو	Nine that with

Figure 52: The Possible Segmentations for Word 'نوجوان'

Any of these can be correct segmentation based on the context words as can be seen in figure 17. Both longest and maximal matching algorithms however, would always identify 'نوجوان' as a single word.

6.2. Statistical Based Technique

Recently, there has been an increasing interest in applying statistical techniques and involving probabilistic models to solve word segmentation predicament. Unlike, the techniques mentioned afore probabilistic word segmentation is based on the context in which word is occurring. The information of neighboring words is often useful to resolve segmentation ambiguities [29, 30]. A few factors are required to be considered when applying probabilistic approach. These are context width and the applied statistical model. The wider the context more is the accuracy, and more is the complexity. As far as statistical model is concerned bi and tri-gram models are more frequently employed.

Another important question to be answered is whether the n-gram model should be applied at character, syllable/ligature or at word level. All the variations have seen to occur in different literatures that talk about probabilistic word segmentation. Viterbi–based technique used by [31] in the initial works of Thai word-segmentation is a character-based technique.

Statistical models can be run on top of rule-based models to apply n-grams on words or smaller clusters/syllables. In most Southeast Asian languages the text can be segmented into smaller units known as clusters or syllables. Unlike word-segmentation, segmenting a text into syllables/clusters is a very easy task. It can be achieved by a finite set of rules. A cluster/syllable is a more well-defined unit than a word because text can be unambiguously segmented into clusters. Work on Lao & Thai syllable segmentation is given in [18] & [34] respectively. Segmenting the word first into syllables effectively removes most of the word-segmentation ambiguities in Thai-text. After segmenting the text into syllables an n-gram model can be applied on syllables.

Syllable based bi-gram model can also be used for Urdu to solve both space-insertion and space-deletion problem. For example sentence ‘نوجوان جواب دو’ (answer me young lad) is first syllabified into ‘نو |جو|ان|جو|اب|دو’ ‘|’ represents syllable boundary. Interestingly each syllable is also a word in this example and possible word segmentations for this sentence are:

نوجوان | جواب | دو	Young lad answer give
نو | جوان | جواب | دو	Nine youngsters answer give
نو | جو | ان | جواب | دو	Nine that them answer give
نوجوان | جو | اب | دو	Young lad that now give
نو | جوان | جو | اب | دو	Nine youngsters that now give
نو | جو | ان | جو | اب | دو	Nine that them that now give

Figure 53: The Possible Segmentations for Sentence ‘نوجوان جواب دو’

Other syllable mergers can be ruled out because they form non-words when combined together. The model based on bi-gram statistics will select the path that maximized the probability which in this case will be:

P(X) =P (نو | s) * P(جو | نو) * P(ان | جو) * P(space | ان) * P(جو | space) * P(اب | جو) * P(space | اب) * P(دو | space) * P(space | دو)

In the later works [32] & [33] variants of tri-gram models are used in conjunction with part of speech trigram model to compute most likely word-segmentation and tag-sequence at the same time. One such model is developed in [35]. A rule based morphological parser JUMAN is used to determine word-segmentation and POS tagging. AMED, a rule based segmentation and POS correction system is then employed. A bi-gram model is finally run to disambiguate the segmentation and POS tagging. The architecture is shown in figure below:

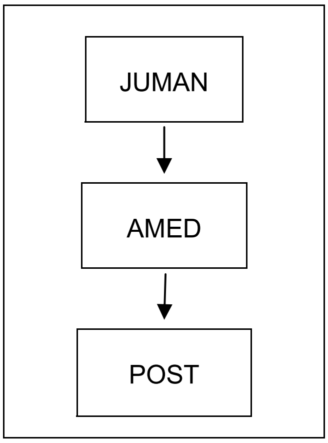


Figure 54: BBN’s JUMAN/AMED/POST Word Segmentation and POS Tagging Architecture

A similar model is also proposed by [36] which use a rule-base to identify 2-character cluster. The algorithm proceeds by sliding a 2-character window over an input sequence and calculating whether this 2-character cluster is a word-boundary or within a word given a previous two character sequence and sequence's status as either word boundary or continuation.

Statistical methods can effectively solve problem of unknown words especially for the constructions like names. Tri-gram model along with part of speech tagging is used as a base by [37] to detect unknown word boundary. A plain trigram model is first used to separate sentence into words. POS is assigned to these words. N-highest probable sequences are than selected. The model is formally defined as:

Let $C = c_1c_2 \ldots c_m$ be an input character sequence. $W_i = w_1w_2 \ldots w_n$ be a possible word segmentation and $T_i = t_1t_2 \ldots t_n$ be a sequence of POS for W_i. Find $W_1W_2 \ldots W_n$ which have N-highest probability of sequence of words. Where:

$$
\begin{aligned}
P\,(W_i) &= \textstyle\sum_t P(W_i\, T_i) \\
&= \textstyle\sum_t \prod_I P(t_i \mid t_{i-1}\, t_{i-2}) \times P(w_i \mid t_i)
\end{aligned}
$$

Where $P(t_i \mid t_{i-1}\, t_{i-2})$ and $P(w_i \mid t_i)$ are computed from the corpus.

Unknown string is detected out of the sequence. It may be a word itself, more then a single word or part of neighboring word. All the possible candidates of unknown words are generated. These are based on following heuristics. For example there is a sentence:

$S = w_1w_2 \ldots w_a\ U\ w_b \ldots w_n$

Where $w_i \in$ Dictionary and $U \notin$ Dictionary

n = Number of Words in the sentence

$UNK = \{X\ U\ Y \mid X \in A \text{ and } Y \in B\}$

Where UNK = set of unknown candidates

$A = \{ w_{a-i,a}\ i \in [0,K]\}\ U\ \{\varepsilon\}$

$B = \{ w_{b,b+i}\ i \in [0,K]\}\ U\ \{\varepsilon\}$

$w_{i,j} = w_{i,\ldots}\ w_{ij}\ i<j$

ε = Null string, K = Constant Value

Figure 55: Equation for Generating Explicit and Partially Hidden Unknown Words

Explicit unknown words are the words that have no sub-string in dictionary for example 'کرسر' (cursor). Partially hidden unknown words are the ones composed of known words and unknown string 'ضرورت مندی' (Need) where 'ضرورت' (Necessity) or 'ضرورت مند' (Needy) might be present in the dictionary but 'ضرورت مندی' itself might not. So it is composed of known + unknown string. In case of fully hidden words both the strings separately exist in dictionary but not in combined form. Example is 'ماں باپ' (Parents) which are not found in dictionaries, although 'ماں' (Mother) and 'باپ' (Father) are both found in dictionaries. This category of unknown words is hardest to detect. The equation for these is given below:

$S = w_1w_2 \ldots w_a \ldots w_b \ldots w_{n-1}w_n$

Where $w_i \in$ Dictionary

n = Number of Words in the sentence

w_a is the word that has probability less than threshold

$UNK = \{X\ U\ Y \mid X \in A \text{ and } Y \in B\}$

Where UNK = set of unknown candidates

$A = \{ w_{a-i,a-1}\ i \in [0,K]\}\ U\ \{\varepsilon\}$

$$B = \{ w_{a+1,a+i}\ i \in [0,K]\} \cup \{\varepsilon\}$$
$$w_{i,j} = w_{i,\ldots}\ w_{ij}\ i<j$$
$$W = w_a : P(w_a \mid t_a) < \text{threshold or}$$
$$W \in w_{a-2,} w_{a-1,} w_a : P(t_a \mid t_{a-1,} t_{a-2}) < \text{threshold or}$$
$$\varepsilon = \text{Null string, } K = \text{Constant Value}$$

Figure 56: Equation for Generating Fully Hidden Unknown Words

If $P(t_a \mid t_{a-1,} t_{a-2})$ < threshold than w_{a-2} and w_{a-1} are also considered to be unknown word because less than threshold probability of w_a might be coming from previous words.

Using these equation unknown candidate words are generated. Based on these candidates new candidate sentences are generated. POS tags are applied to these sentences through trigram statistics. Unknown words are given proper noun tags. Let W be a sequence of words $w_1 \ldots w_n$ and T_i be a sequence of tags $t_1 \ldots t_n$. Find P(X) which maximizes $P(T_i \mid W)$:

$$\begin{aligned} P(X) &= \text{argmax}_{T_i} P(T_i \mid W) \\ &= \text{argmax}_{T_i} P(t_i \mid t_{i-1}\ t_{i-2}) \times P(w_i \mid t_i) \end{aligned}$$

There are some problems with n-gram statistics. First it considers only coarse information of part of speech in a fix restricted range of context. Long distance dependencies and word collocations may be easily ignored and some important information might be lost [38].

Another problem with this technique is that results are heavily dependant on a segmented training corpus. It requires enormous training corpus to estimate all the parameters correctly. Corpus preparation is a very time consuming and laborious task. And yet another problem is that too many functional (or close class words) can make the analysis biased.

Nevertheless, statistical methods are still assumed to be very effective and proven technique to solve for both segmentation ambiguities and unknown word problem. They can be used standalone or in merger with rule-based techniques accompanied.

6.3. Feature-Based Approach

Feature based techniques are used to overcome the shortcomings of statistical techniques. A feature can be anything that tests for specific information in the context around the target word sequence, such as context words and collocations. Instead of using one type of syntactic evidence as in N-gram approaches, we can apply synergy of several types of features. The idea is to learn several sources of features that characterize the context in which each word tends to occur. Then these features are combined to remove the segmentation ambiguities [37].

Context words feature tests for a presence of a particular word within +/- K words of the target word. Collocation tests for a pattern of up to L contiguous words and POS around the target word. All the possible ambiguous strings in training corpus are registered as special entries in dictionary. Confusion set is generated by listing all the possible segmentations. For example confusion set for entry ‘มากว่า’ is {มา กว่า, มาก ว่า}. Then features are learned for each element of the confusion set. Example of feature for the above given confusion set include [38]:

- ฝูค within -10 Words
- มา กว่า Collocation

The first feature uses context word to disambiguate and prefers 'มาก ว่า'. Second uses character collocation and implies 'มา กว่า'.

Syllable based collocation is also implied by [39]. It views word segmentation as two step process. First segmenting the text into syllables and then merging the syllables. The idea is that collocation strength between two syllables that are part of a word is more than collocation that are not part of word. For example in a sequence of syllables …a-b-c-d-e… in which 'b-c-d' is a word the collocation strength between 'b-c' and 'c-d' is more than the collocation strength between 'a-b' and 'd-e'. The overall collocation of a sentence can be defined as:

$$
\begin{aligned}
&St=\sum_{i=1}^{n} F_{w_i}-\sum_{i=1}^{n-1} D_{w_i,w_{i+1}}\\
&F_{w_i}=\sum_{j=1}^{k-1} C_{s_j,s_{j+1}} \text{ such that } w_i=s_1s_2\ldots\ldots s_k\\
&D_{w_i,w_{i+1}}=C_{s_j,s_{j+1}}\\
&\text{such that } s_j \text{ is the last syllable of } w_i\\
&s_{j+1} \text{ is the first syllable of } w_{i+1}
\end{aligned}
$$

Figure 57: Over-all Collocation of a Sentence

Collocation between two syllables x-y can be defined as [39]:

$$
\begin{aligned}
&\log\frac{p(x,y)}{q(x,y)}=\log\frac{p(x)p(y\mid x)}{q(x)q(y\mid x)}=\log\frac{p(y\mid x)}{q(y\mid x)}\\
&=\log\frac{Count(x,y)/Count(x)}{Count(x,Any,Y)/Count(x)}\\
&=\log\frac{Count(x,y)}{Count(x,Any,y)}
\end{aligned}
$$

Figure 58: Colloaction between Two Syllables

Where p(x, y) is probability of finding syllables x and y together and q(x, y) is the probability of finding any syllable between x and y.

Another feature generates all the possible prefix sets. A prefix set is a set of words where 'a' and 'b' are two words in set and either 'a' is a prefix of 'b' or 'b' is a prefix of 'a'. An example of prefix set is {มา, มาก, มากมาย}. {มา, มาก} is another prefix set which we must consider too. Based on the feature set training examples are extracted from the corpus which are used to extract features. These features are than used to decide whether 'มากมาย' is segmented into 'มา กมาย', 'มาก มาย' or 'มากมาย'.

Two popularly used machine learning techniques Winnow and Ripper are commonly used to extract discriminative features from the feature space. Given below is brief introduction about these.

6.3.1. Winnow

Winnow forms a neural like network in which target node is connected to several nodes. Each of these nodes is a specialist that looks at a particular value of an attribute of the target concept, and will vote for a value of the target concept based on its specialty; i.e. based on the value of attribute it examines. The global algorithm will then decide on weighted majority votes receiving from those specialists. The pair (attribute=value) that a specialist examines is a candidate of features we are trying to extract [40].

6.3.2. RIPPER

RIPPER is a propositional rule learning algorithm that constructs a rule set which accurately classifies the training data. A rule in constructed rule set is represented in the form of conjunction of conditions:

If T_1 and T_2 and … T_N then C_X

T_1 and T_2 and … T_N is called the body of the rule. C_X is the target class to be learned; it can be positive or negative one class problem, or any class in case of learning multiple classes. A condition T_i tests for a particular value of an attribute, and takes one of four forms: A_N=v, $A_c \geq \Omega$, $A_c \leq \Omega$ and $v \in A_s$ where A_N is nominal attribute and v is a legal value of $A_{N;}$ A_c is a continuous variable and Ω is some value of A_c that occurs in the training data.

Prime focus of feature based techniques is to remove ambiguities in text. These run on top of rule based or statistical techniques. General model is given shown in figure below:

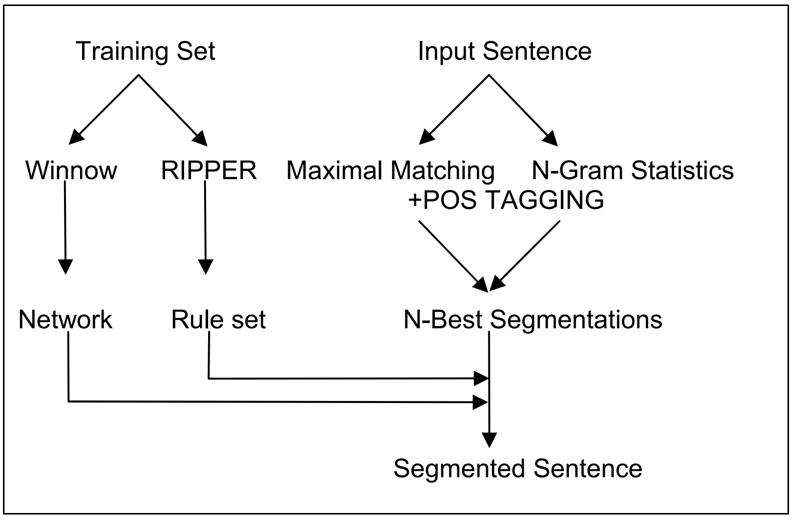


Figure 59: Feature Based Segmentation System [38][5]

7. Methodology

Methodology adopted for accomplishment of our word segmentation model defined in section 4 include following steps.

1. Studying of segmentation errors in Urdu
2. Data collection
3. Implementing segmentation model for Urdu
4. Algorithm

[5] This figure is a modified version of [38]

7.1. Studying Word Segmentation Trends in Urdu

A study was performed to identify segmentation problems in Urdu. The data used for analysis was taken from BBC and Jang corpus mainly. A data of 5,000 words from both corpuses was looked at. The percentage for each problem was calculated. Given below are the results of study:

Table 8: Error Statistics from a Study of BBC and Jang Corpus of 5000 Words each

	BBC Corpus		Jang Corpus		Total	
Space Insertion	373	7.46%	563	11.26%	936	9.36%
Affixation	298	3.96%	467	9.34%	765	7.65%
Reduplication	52	1.04%	76	1.52%	128	1.28%
Compounding	133	2.66%	218	4.36%	351	3.51%
Abbreviations	263	5.26%	199	3.98%	462	4.62%

Table 9: Percentage of Total Errors

	Number of Errors	%age of Total Errors
Space Insertion	936	35.42%
Affixation	765	28.95%
Reduplication	128	4.84%
Compounding	351	13.28%
Abbreviations	462	17.48%

7.2. Collecting Data

This step involved collecting data to be used for the model. A list complete list of 102863 “words[6]” was obtained from CRULP resources. This list contained 53513 common words and 49350 proper nouns. This list also contained compound words, words with affixes, company names, and foreign English words. These words are also POS tagged as nouns, adjective, verbs, auxiliary, adverb, pronouns and some other tags for functional words.

7.2.1. Collecting Free Morphemes

This list was cleaned up so as to extract free morphemes. These included common words with no affixes. Proper names of cities and countries were separated and added to list. Only the names that did not contain any spaces were chosen. Person names were also extracted and added to this list of morphemes. After cleaning the data a list of approximately 62,000 free morphemes were obtained. POS tags for these morphemes were also extracted. POS tags are used in affixation module discussed later.

7.2.2. Data Collection

A list of approximately 17,800 names was also extracted from a word list obtained from CRULP resources. These are kept separate and are use in abbreviation module discussed later.

7.2.3. Collecting Compound List

[6] Not the word defined in section 4.

While extracting a list of free morphemes Urdu compound words were also separated and a list of approximately 1850 compound words was extracted. This list is used as a look for compound words.

7.2.4. Collecting List of Affixes

A comprehensive list of 450 affixes was obtained from a side project in CRULP. 20 others were added during the phase of testing. These affixes were divided into 2 categories prefixes and suffixes. This altogether formed a list of 60 prefixes and 410 suffixes.

In Urdu some affixes can occur in both in free and bound forms. When they occur in free form they are not part of the words but hold status of words. Each of these two was further categorized into free and bound prefixes and suffixes. Given below are the results

Table 10: Free and Bound Affix Status

Prefixes		Suffixes	
Free	Bound	Free	Bound
56	4	197	213

A comprehensive list of bound and free prefixes and suffixes is given under Appendix G.

Separating free and bound affixes helps joining morphemes in affixation module. Bound affixes can be joined with preceding or following word immediately. Free affixes are required to be examined carefully before joining. This is further discussed in affixation module.

7.2.5. Collecting Unigram and Bigram Frequencies

Against each of the words collected from corpus a list of unigram frequencies was obtained from CRULP resources. Unigram frequencies for proper names was not available '1' was assigned to each of such word. Each of the normalized frequency was divided by 18308616 (the grand total of all the frequencies) to obtain unigram probabilities.

A list of 36393 bi-grams probabilities were obtained from another work, a side thesis "POS tagger". These bi-grams probabilities are obtained from a BBC tagged corpus of 80,000 words.

7.2.6. Collecting Space Insertion Instances with Spelling Variations

A list of space insertion problems that exhibit spelling variation was extracted from the study of BBC and Jang corpus. A list of 36 such cases was extracted. Most of these are given in Appendices.

7.3. Components of Segmentation Model

This section explains the implantation details for implementing model shown in figure 51, section 4. Each of the problems is solved in a different module and all the modules merge together to give final output. All the modules are discussed one by one and then merge details are given. The sequence of writing is not as they fit into model or as the algorithm moves. These should be read as separate component. Section 7.4 gives algorithmic details and description how these modules jell together.

7.3.1. Maximum Matching Module

Maximum matching module is given a string without any spaces. The function of this module is to get all possible segmentations from the input string and rank them. Maximum matching module goes through following steps.

7.3.1.1. Extracting Words

For a given input extract all possible valid words from list of free morphemes (we call it corpus from now onwards). For example on input string 'سرمدحسین' maximum matching module extracts 10 possible words given as ' سرمد', 'سر', ' رمد', 'رم', ' مدح', ' مد', ' حسین','حسی', 'سین' and 'سی'. These ten words act as our mini data base for generating segmentations.

7.3.1.2. Generating Segmentations

The module now generates segmentations from the extracted words. For each of the words extracted the module now finds which of these are starting of the input sentence and store each of these separately. In the given example 'سرمد' and 'سر' are found. Each of the starting words is stored in two dimensional structure as shown below

سرمد						
سر						
Empty Slot						

Figure 60: 2-D Matrix-Generating Segmentations

Now for each of the starting word a next possible word is chosen from the list by looking at input sequence. For example next possible word for 'سرمد' is 'حسین' and 'حسی' where as next possible word for ''سر' is 'مد' and 'مدح'. So we put these options and fill the 2-D matrix. If a starting point has more than one option then the entire branch is copied onto Empty Slot and the variable is progressed to next index. After this iteration the matrix looks like

سرمد	حسین					
سر	مدح					
سرمد	حسی					
سر	مد					
Empty Slot						

Figure 61: 2-D Matrix-Generating Segmentations

Now we have four starting points namely 'حسین', 'مد', 'حسی' and 'مدح'. For each of these staring points we again search for next expected word branch when required. If no matching entry is found against expected entry for a starting point then either the string has ended or no such word exists in words extracted in step 7.3.1.1. In this case that letter is dropped, error is reported and searched is started from next character. For example in above given when searching next entry for 'حسی' the program tries to look for words starting from 'ن' no such entry is found error is reported 'ن' is added to segmentation matrix. Similarly when searching next entry for 'مدح' the program tries to look for words staring from 'ی' no such entry is found so it registers an error and tries to look words staring from ‘ن’. After all the starting points have been completely exhausted the 2-D matrix will look as shown in figure below:

سرمد	حسین					
سر	مدح	سین				
سرمد	حسی	ن				
سر	مد	حس	ن			
سرمد	مد	حسین				
سر	مدح	سی	ن			
سر	مد	حسی	ن			
سر	مد	حس	ی	ن		
Empty Slot						

Figure 62: 2-D Matrix-Generating Segmentations

When a word is chosen as a potential next possible word to a current starting point it is verified against the input string. For example if a starting point is ABC and the expected next entry starts from words D. The program selects all the entries starting from D from mini data base but verifies each of these against the input string. Let say the input string is “DTOABCDEF”, the mini database has ‘DTO’, ‘DE’ and ‘DEF’. At starting point ‘BC’ the expected entry should start from ‘D’ potential candidates are ‘DTO’, DE’ and ‘DEF’. But the program will reject ‘DTO’ by verifying it against input string. The program maintains an index on where in input string it is.

Against each of the generated segmentation its word and error count are also maintained. So 2-D matrix looks like:

سرمد	حسین					2	0
سر	مدح	سین				3	0
سرمد	حسی	ن				3	1
سر	مد	حس	ن			4	1
سرمد	مد	حسین				3	0
سر	مدح	سی	ن			4	1
سر	مد	حسی	ن			4	1
سر	مد	حس	ی	ن		5	2
Empty Slot							

Figure 63: 2-D Matrix-Generating Segmentations

The program also maintains which of the elements in mini database are used in generating segmentations. There might be a possibility that we have missed out some segmentation. For example in above given mini database two elements ‘رمد’, and ‘رم’ are not used in any segmentations. The program now generates segmentations against each of these. For a missed out entry ‘رمد’ the program detects its initial and final string from the input string. These are ‘س’ and ‘حسین’. The program recursively calls generating segmentation step discussed in this section to generate all possible segmentations for each of the preceding and following strings. After all possible segmentations are found these are merged together. For example against ‘رمد’ preceding string generates only one segmentation ‘س’ where as following string ‘حسین’ generates three segmentations namely ‘حسین’, ‘حسی’ + ‘ن’ and ‘حس’, ‘ی’ and ‘ن’. If ‘n’ is the number of segmentations obtained from preceding string and ‘m’ is the number of strings obtained from following string then m x n new segmentations are obtained. Similar procedure is repeated for untouched entry ‘رم’ where segmentations for ‘س’ and ‘دحسین’ are generated and merged. The final 2-D matrix that we devised now updated as shown in figure below:

سرمد	حسین					2	0
سر	مدح	سین				3	0
سرمد	حسی	ن				3	1
سر	مد	حس	ن			4	1
سرمد	مد	حسین				3	0
سر	مدح	سی	ن			4	1
سر	مد	حسی	ن			4	1
سر	مد	حس	ی	ن		5	2
س	رمد	حسین				3	1
س	رمد	حسی	ن			4	1
س	رمد	حس	ی	ن		5	2
س	رم	د	حسین			4	1
س	رم	د	حسی	ن		5	2
س	رم	د	حس	ی	ن	6	3
Empty Slot							

Figure 64: 2-D Matrix-Generating Segmentations

7.3.1.3. Selecting Best Segmentations

After all the segmentations have been generated the program now selects best segmentation. The selection is based on minimum number of words heuristic. Segmentations are sorted based on minimum number of words. If two segmentations have same number of words then their error count is compared. 10 best segmentations are selected from the lot.

7.3.2. Handling Space Insertion Instances with Spelling Variations

Maximum matching module handles most of the space insertion problems however the ones with spelling variation are not taken care of. These are dealt in a different module. Given a string 'کیخلاف' maximum matching module will output 'کی' and "خلاف which is not the right segmentation. Such variations can be affectively dealt with running a spell checker on a given input before sending the input to maximum matching module. As already mentioned, during study of BBC and Jang corpus a list of such problems was extracted. This list is complete by no means but it covers most common occurrences of this problem. For example 'کے' and 'لیے' when come together are often written in a joined fashion as 'کیلیے'.

This module is run over maximum matching module. If 'کیلیے' or any such instance registered in the list occurs it is broken into its proper constituents. The status of both of the constituents is now "resolved" so these are not send to maximum matching module.

Some of such instances like 'دیدی' and 'لیکر' etc are also free words or part of words in Urdu so such instances are not broken into constituents and sent to maximum matching module to analyze.

7.3.3. Handling Reduplication

The instances retrieved from BBC and Jang data showed that Urdu exhibits both full and partial reduplication. Partial reduplication further had patterns (already discussed in section 2.4). Giving a deeper insight into all the instances collected from corpus one finds out that they share a property. In a reduplication instance XY Y is either X or a variation of X. In case of later changing Y by a character or two can get us X or vice a versa. Therefore given an input sequence 'a_1, a_2,

a_3, a_4…a_n. edit distance algorithm can be applied on each a_{x-1}, a_x. If the edit distance between these two is 1 then these two can be merged into a single unit under the tag of reduplication.

The case of full reduplication is trivial and can be done simply comparing if a_{x-1} and a_x are equal. Reduplication test is only applied if length of a_{x-1} and a_x is greater then 3. The reason for this is that there is a great probability that words of length smaller than 4 might not be instances of reduplication but the edit distance between them is 1. One such instance is 'کیا گیا' that repeatedly occurs in corpus and has edit distance 1. But marking 4 as a cut off point also leaves examples like 'ڈیل ڈول' and ' الگ تھلگ' as two separate entities.

Reduplication is also run over maximum matching module. The reason for that is that the reduplicated word Y is often not present in the word list. So maximum matching module will break 'Y' into some other segments after which it would be impossible to join it back. As an example 'تھلگ' if send to maximum matching is broken into 'تھل' and 'گ'.

Reduplication check is also applied on single word Z because there might be a case that Z contains XY where X ends with a non-joiner character. One such example is 'ڈھیلاڈھالا' where X='ڈھیلا' and Y='ڈھالا'. To examine such cases Z is split into equal two halves and edit distance is applied on the two halves. If it results in 1 the two are registered as reduplication and not sent to maximum matching approach.

Single edit distance algorithm is given in Appendix H.

7.3.4. Handling Affixation

As already mentioned affixes are identified as free and bound morphemes. The purpose for identifying free morpheme affixes was to conduct an analysis. The idea used is to gather all the words in list (obtained from NOKIA) where that affix is used. Separate affix and the root morphemes from each of the searched words. The POS for all the roots were extracted. From these POS we get information that what are the most common POS's that get connected with this affix. For example against suffix 'ناک' we found of 30 words from corpus. The POS's for the root of each of these words were extracted. Out of 30 instances 28 were NOM (Nouns), 1 was found to be ADJ (Adjective) and 1 of these was ADV (Verb). Therefore it can be inferred that suffix 'ناک' gets connected with preceding word when it is a noun. Similarly 352 word occurrences for prefix 'غیر' were found. These were analyzed to infer that 'غیر' comes as a prefix before words NOM, ADJ and ADV. A small VB program was written to obtain these results.

Having analyzed affix status for all the free morpheme affixes we are now ready to join free morphemes obtained from maximum matching module. A 2-D matrix similar to the one shown in section 7.3.1 is passed to affixation handler. Every morpheme 'X' in 2-D matrix is checked whether it is an affix. If a morpheme is found to be an affix then it is checked whether it is a prefix or suffix and whether it is a free or bound affix. If it is a non-free affix it is joined immediately with preceding or following morpheme depending upon if it is a prefix or suffix. If it is a free morpheme affix then POS for preceding or following morpheme is extracted. If the POS of that morpheme matches with POS status of this affix then the two are combined. Otherwise they are not combined. For example in following two examples affixation handler operates as shown under.

اس	کا	انجام	عبرت	ناک	ہو	گا

Figure 65: Affix Handler - Example 1

For suffix 'ناک', POS for 'عبرت' is looked at. Because POS of 'عبرت' is NOM which matches with the POS status of 'ناک' these two are combined. On contrary in following

لمبی	ناک	والے

Figure 66: Affix Handler - Example 2

will not join suffix 'ناک' with because POS for 'لمبی ' is ADJ which does not match with POS status of 'ناک'.

This technique is not fool proof. It does is able to correctly detect whether an affix is occurring as a free morpheme or as an affix in a sentence in most of the cases but fails in some case. For example the model will incorrectly join 'بیٹا' and 'ناک' in following example.

بیٹا	ناک	صاف	کرو

Figure 67: Affix Handler - Example 3

because POS of 'بیٹا' is also NOM. Another such example is

وہ	فراری	کار	چلانا	چاہتے	ہیں

Figure 68: Affix Handler - Example 4

Where program incorrectly identifies 'کار' as a suffix and combines it with 'فراری'. Because suffix 'کار' combines with root words that has POS noun.

This technique can be improved by improving POS tag set. 'بیٹا', 'فراری' and 'عبرت' and 'تخریب' are obviously nouns but if nouns are further classified into abstract nouns and other nouns are separated these errors can be tackled.

However, there would still be errors that can not be handled using this technique. For example consider following two examples:

وہ	اس	بات	پر	سکون	محسوس	کرتے	ہیں
وہ	بہت	پر	سکون	نظر	آتے	ہیں	

Figure 69: Affix Handler - Example 5

In both example 'پر' precedes 'سکون'. However, in first example 'پر' is postposition or case marker where as in second example it is a prefix. The technique used in this thesis will not be able to differentiate between the two. This problem can be solved using n-gram based statistical analysis. Also if the data was available in diacritized form this problem could be solve because postposition 'پر' has /a/ short vowel sound where as prefix 'پر' has /u/ short vowel sound.
Because 'پر' and 'ان' are more commonly used as free morphemes and not as affixes in corpus and the technique employed here does not accurately identifies whether these are prefix or free morpheme we have removed these from the list of affixes.

This module also tries to handle multiple affixations. If there is already a prefix attached to a morpheme and a suffix proceeds, the suffix will also be joined to form a word with prefix and a suffix.

7.3.5. Abbreviation Handler

This module detects abbreviations in input text and tries to merge them if they are not preceding a name. From the BBC and Jang corpus analysis it was found out that abbreviation when occurring in names follow 'X' 'X' 'Name' format where 'X' is pronunciation of an English letter. A model was designed based on this heuristic.

This module is run on results of maximum matching on 2-D matrix similar to the one in previous sections. For each morpheme it is checked whether it is an English letter. If that is the case its preceding morpheme is checked. If preceding morpheme is also an English letter following morpheme is checked. If following morpheme is not a name then merge preceding and this morpheme and put them into cell. If preceding cell contains already merged English letters merge this morpheme with previous cell. Example below illustrates the functionality.

پی	ائی	اے	پرواز	پی	کے	786	
میں	آر	ڈی	برمن	کے	گانے	سنتا	ہوں

Figure 70: Abbreviation Handler - Example 1

In the first example when looking at morpheme 'ائی' previous morpheme is checked 'پی' is also an English character so following morpheme is checked it is not name so 'پی' and 'ائی' are merged. The position of 2-D array now is:

پی ائی	اے	پرواز	پی	کے	786

Figure 71: Abbreviation Handler - Example 2

When repeating this procedure for morpheme 'اے' similar test is conducted when all fail it checks whether preceding cell contains an abbreviation it will be merged into previous cell to give output:

پی ائی اے	پرواز	پی	کے	786

Figure 72: Abbreviation Handler- Example 3

On contrary in second example 'آر' and 'ڈی' are not combined because a name 'برمن' follows. Because 'کے' is very commonly used case marker in Urdu there might be a possibility that it is incorrectly identified as abbreviation. For example in this string 'ایم کیو ایم کے نمائندے' 'کے' is not part of abbreviation. For that matter we do not merge 'کے' when it occurs as an ending letter in abbreviation. However, if another letter follows it we merge it. There fore 'ایم کے ایم' will be merged.

Merging morphemes in this fashion is not fool proof. During testing it was found that a name can follow even after three abbreviation morphemes. For example in following instance:

اے پی جے عبدالکلام (A.P.J Abdul Kalam) name follows after three letters. Also there might be a possibility that in a sentence two different abbreviation sets occur consecutively. For example

a	میرا	پی	سی	اے	سی	کے	سامنے	ہے
b	میرا	پی سی اے سی	کے	سامنے	ہے			
c	میرا	پی سی	اے سی	کے	سامنے	ہے		

Figure 73 (a) Input Sequence, (b) Erroneously Detected Segmentation, (c) Correct Segmentation

In this case our program erroneously assumes these to be one abbreviation and merge them, which is not the case as (c) is the correct segmentation.

7.3.6. Compound Handler

Compound handler module takes a 2-D matrix obtained from maximum matching module. For each morpheme it tries to combine following 'n' morphemes (where n is 4) and performs a lookup in the compound list of 1850 compounds. If match is found it combines the next 'n' morphemes and merge them to starting morpheme otherwise the test is applied to 'n-1' morphemes and so on. If no match is found up till ith morpheme (where i is staring point) then i is progressed ahead by 1 index. This technique is known as longest matching. Given below is an example:

پاکستان	کے	وزیر	اعظم	شوکت	عزیز	نے	کہا

Figure 74: Compound Handler

Let say 'i' is currently 3 that is at morpheme 'وزیر'. It will try to merge next 4 morphemes to form a search entry 'وزیر اعظم شوکت عزیز نے کہا' and searched into compound data base. The entry is not found so 'n' is reduced to 3 and new search entry is 'وزیر اعظم شوکت عزیز نے'. This entry is not found when 'n' is 1 the search entry is 'وزیر اعظم' which is found in compound data base so these two are merged.

7.3.7. Removing Diacritization

This module removes diacritics from the input data. Diacritics can be helpful in compounds with zer-e-izafat and help to find out if ‘پر’ is a prefix or a postposition. However use of diacritics is not very common. These are only used for beautification in some text. Also if input with diacritics is allowed the entire training data must also be diacrtized which is extremely inefficient. Therefore all the diacritics are removed at the start.

7.3.8. Orthographic Word Separator

Function of this module is to extract orthographic words from a given input. Orthographic words are defined as sequence of character separated by spaces. No of orthographic words in an input is one greater than number of spaces it has. For example ‘میں نےدل سےکہا دھونڈلا ناخوشی’ has 4 white space characters and 5 orthographic words. As we know white space does not indicate word boundary in all cases but in more than 50% cases it does. If we remove spaces from our input we are loosing some information. Space gives a division point to word segmentation problem. More the number of spaces in a sentence lesser will be segmentation possibilities longer a string gets more are the segmentation possibilities. Let us say X and Y are two separate sequences of character. Let us say there are m possible segmentations of X and n are the possible segmentations of sequence ‘Y’. Merging segmentations of X and Y we get m x n segmentations. However if there is no space between X and Y and let Z be that sequence of characters that generate p segmentations then p will be greater then m x n and it would be difficult to select best segmentation. This is because maximum matching module operates in exponential manner bigger the string more will be the possibilities. Having space, rule out some of the options that are impossible. Therefore removing space means program is unnecessarily generating extra segmentations and then doing extra work to choose best out so many never intended segmentations. For example we get an input string ‘نادر خان درانی’ and send each of the orthographic word we get 2 x 1 x 7=14 segmentations. However total number of possible segmentations obtained for ‘نادرخاندرانی’ is 77, almost 5 times bigger. Imagine how big this number can get with 10-15 word sentences. It is much easier and efficient approach to select best segmentation from 14 sentences then from 77.

7.3.9. Segmentation Mergering

Segmentation merge module combines segementations obtained from maximum matching module against each orthographic word. If m is the number of segmentations obtained from orthographic word X and n are the segmentations obtained from orthographic word Y then total number of segmentations for X Y will be mxn. Segmentations are ranked by min word min error heuristic after each merge. Also if the segmentation count goes beyond 50 only top 50 are selected. When X Y are merged these are put together in Z. For next merge Z becomes X and segmentations obtained from next orthographic word are put in Y. Let us demonstrate this below by taking an example: ‘میرےبعد کسکو ستاؤگے’. There are three orthographic words, ‘میرےبعد’, ‘کسکو’ and ‘ستاؤگے’. After maximum matching module possible segmentations for each of these are obtained:

<table>
<tr><td>میرےبعد</td><td colspan="2">کسکو</td><td>ستاؤگے</td></tr>
<tr><td>میرے|بعد</td><td rowspan="3">کس</td><td rowspan="3">کو</td><td>ستاؤ|گے</td></tr>
<tr><td>می|رے|بعد</td><td>ستا|ؤ|گے</td></tr>
<tr><td>میرا|ے|بعد</td><td>ست|اؤ|گے</td></tr>
<tr><td></td><td></td><td></td><td>س|تاؤ|گے</td></tr>
<tr><td></td><td></td><td></td><td>س|تا|ؤ|گے</td></tr>
</table>

Figure 75: Segmentations Obtained Against Each Orthographic Word

‘کسکو’ is not sent to maximum matching module it is dealt by spelling variation module. After first two merges the above figure is like

<table>
<tr><th>X</th><th>Y</th></tr>
<tr><td>میرےبعد</td><td>ستاؤگے</td></tr>
<tr><td>میرے|بعد|کس|کو</td><td>ستاؤ|گے</td></tr>
<tr><td>می|رے|بعد|کس|کو</td><td>ستا|ؤ|گے</td></tr>
<tr><td>میرا|ے|بعد|کس|کو</td><td>ست|اؤ|گے</td></tr>
<tr><td></td><td>س|تاؤ|گے</td></tr>
<tr><td></td><td>س|تا|ؤ|گے</td></tr>
</table>

Figure 76: First Three Segmentations Merged

After final merge Z has 15 segmentations shown in figure below:

<table>
<tr><td>| یرے | بعد | کس | کو | ستاؤ | گے</td></tr>
<tr><td>| می | رے | بعد | کس | کو | ستاؤ | گے</td></tr>
<tr><td>| میرے | بعد | کس | کو | ستا | ؤ | گے</td></tr>
<tr><td>| میرے | بعد | کس | کو | ست | اؤ | گے</td></tr>
<tr><td>| میرے | بعد | کس | کو | س | تاؤ | گے</td></tr>
<tr><td>| میر | ے | بعد | کس | کو | ستاؤ | گے</td></tr>
<tr><td>| می | رے | بعد | کس | کو | ستا | ؤ | گے</td></tr>
<tr><td>| می | رے | بعد | کس | کو | ست | اؤ | گے</td></tr>
<tr><td>| می | رے | بعد | کس | کو | س | تاؤ | گے</td></tr>
<tr><td>| میرے | بعد | کس | کو | س | تا | ؤ | گے</td></tr>
<tr><td>| میر | ے | بعد | کس | کو | ستا | ؤ | گے</td></tr>
<tr><td>| میر | ے | بعد | کس | کو | ست | اؤ | گے</td></tr>
<tr><td>| میر | ے | بعد | کس | کو | س | تاؤ | گے</td></tr>
<tr><td>| می | رے | بعد | کس | کو | س | تا | ؤ | گے</td></tr>
<tr><td>| میر | ے | بعد | کس | کو | س | تا | ؤ | گے</td></tr>
</table>

Figure 77: Finally Merged Segmentation

The segmentation count doesn't go beyond 50 so no segmentation is ruled off.

7.3.10. Ranking Segmentations

Segmentations are ranked many times during entire course of algorithm. This section discusses three different techniques that we have used to rank segmentations.

7.3.10.1. Minimum Word – Minimum Error Heuristic

This heuristic selects the best segmentation based on number of words it has and selects the one with lesser number of words. If segmentations have equal number of words the one with lesser number of errors is selected. If error count for these segmentations is also equal then the one at first index of 2-D array is selected.

This technique works pretty well in most of the scenarios but give incorrect segmentations in many cases. For example best segmentations generated by input ' اس نے پرفارم کرنے کافیصلہ کیا ہے' have 8 words each and there are four such segmentations. All of these have zero error. These are given below as the program generates them:

<table>
<tr><td>|اس |نے |پرفارم |کرنے |کافی |صلہ |کیا |ہے |</td></tr>
<tr><td>|اس |نے |پرفارم |کرنے |کافی |صلہ |کی |اہے |</td></tr>
<tr><td>|اس |نے |پرفارم |کرنے |کا |فیصلہ |کیا |ہے |</td></tr>
<tr><td>|اس |نے |پرفارم |کرنے |کا |فیصلہ |کی |اہے |</td></tr>
</table>

Figure 78: Max Matching Segmentation Options

All of these have equal number of words i.e. 8 and equal numbers of errors i.e. 0. This technique always selects the first out of lot which gives incorrect segmentation. The correct one is at 3rd number. Consider another example 'آپکانام کیاہے'. There are eight such segmentations with 5 words and 0 errors as shown below

<table>
<tr><td>آپ | کان | ام | کیا | ہے |</td></tr>
<tr><td>آپ | کان | ام | کی | اہے|</td></tr>
<tr><td>آپ | کا | نام | کیا | ہے|</td></tr>
<tr><td>آپ | کا | نام | کی | اہے|</td></tr>
<tr><td>آ | پکا | نام | کیا | ہے|</td></tr>
<tr><td>آ | پکا | نام | کی | اہے|</td></tr>
<tr><td>آ | پک | انام | کیا | ہے|</td></tr>
<tr><td>آ | پک | انام | کی | اہے|</td></tr>
</table>

Figure 79: Maximum Matching Segmentation Options

7.3.10.2. Unigram Based Selection

As already discussed maximum matching approach does not have any clue when more then one segmentations with equal number of words and errors occur. It arbitrarily chooses first one which is not always the correct segmentation. Unigram based technique gets unigram probability (from corpus) for each morpheme in segmentation and multiplies all the frequencies. Best segmentation is the one with higher cumulative frequency.

Let C = $c_1c_2…c_m$ be an input character sequence. S_i = $m_1m_2…m_n$ be a possible morpheme segmentation. Find S_x which have highest probability of sequence of morphemes. Where:

$$P\ (S_x) = \text{Argmax}\ (P\ (S_i))$$
$$P\ (S_i) = \prod_i P\ (m_i)$$

Where P (m_i) is computed from the corpus.

For example unigram frequency count for each of these segmentations in above example is given as following:

0.0133 x 0.0138 x 2.84 x e^{-5} x 0.03527 x 2.85 x e^{-4} x 1.19 x e^{-4} x 0.0592 x 0.0255 = 9.5927464732258365E-24	اس نے پرفارم کرنے کافی صلہ کیا ہے
0.0133 x 0.0138 x 2.84 x e^{-5} x 0.03527 x 2.85 x e^{-4} x 1.19 x e^{-4} x 0.0314 x 1.58 x e^{-6} =3.2720704203424786E-27	اس نے پرفارم کرنے کافی صلہ کی اہے
0.0133 x 0.0138 x 2.84 x e^{-5} x 0.03527 x 0.0167 x 4.65 x e^{-4} x 0.0592 x 0.0255 = 2.1912770766460559E-20	اس نے پرفارم کرنے کا فیصلہ کیا ہے
0.0133 x 0.0138 x 2.84 x e^{-5} x 0.03527 x 0.0167 x 4.65 x e^{-4} x 0.0314 x 1.58 x e^{-6} = 7.4744109262974958E-24	اس نے پرفارم کرنے کا فیصلہ کی اہے

Figure 80: Unigram Frequencies

Unigram frequencies for all 50 segmentations are calculated and best segmentation is selected. In some cases minimum word heuristic is not correct. Consider following example:

<table>
<tr><td>Input Sentence</td><td>’انکی پیدا کردہ دشواریوں کےباوجود‘</td></tr>
<tr><td>Max Match</td><td>انکی|پیدا کردہ|دشواریوں|کے|باوجود|</td></tr>
<tr><td>Unigram</td><td>|ان|کی|پیدا کردہ|دشواریوں|کے|باوجود|</td></tr>
</table>

Figure 81: Unigram Vs Max Matching

As can be seen that segmentation with more number of words is the correct one and unigram correctly ranks it higher. Minimum word heuristic fails in this case.

However, there is a problem with unigram method. The context window i.e. is one word is too small. Unigram frequencies for functional words are very high because of which segmentation with functional words is best ranked in some case. Consider following example where maximum word technique works better.

<table>
<tr><td>Input Sentence</td><td>سونیااسکی آیڈیل ہے</td></tr>
<tr><td>Max Match</td><td>|سونیا|اس|کی|ائی|ڈیل|ہے</td></tr>
<tr><td>Unigram</td><td>|سو|نیا|اس|کی|ائی|ڈیل|ہے</td></tr>
</table>

Figure 82: Unigram Vs Max Matching

Another such example is

<table>
<tr><td>Input Sentence</td><td>بہت ساری اسکیمیں ہیں</td></tr>
<tr><td>Max Match</td><td>|بہت|ساری|اسکیمیں|ہیں|</td></tr>
<tr><td>Unigram</td><td>|بہت|ساری|اس|کی|میں|</td></tr>
</table>

Figure 83: Unigram Vs Max Matching

7.3.10.3. Bigram Based Selection

Bigram technique tries to improve on the drawback of unigram. Major problem with unigram is that it does not look at context. Bigram looks at neighboring morphemes and decides based on that. Bigram method evaluates probability of a morpheme given previous morpheme. Given a word pair XY, calculate all its occurrences from the corpus. Let this number be ‘n’. Count all the occurrences of X from the corpus. Let this number be ‘p’. Bigram probability for word pair XY is n/p. For an input sequence of morphemes m_1, m_2, m_3, m_4….m_n bi-gram frequency for each pair $m_{j-1,}$ m_j is calculated. For starting morpheme $m_{1,}$ m_1|Start (m_1 given start) is calculated. These frequencies are multiplied to obtain cumulative probability.

Let C = $c_1c_2…c_m$ be an input character sequence. S_i = $m_1m_2…m_n$ be a possible morpheme segmentation. Find S_x which have highest probability of sequence of morphemes. Where:

$$P\ (S_x) = \text{Argmax}\ (P\ (S_i))$$
$$P\ (S_i)\ = \prod_i P\ (m_i \mid m_{i-1})$$

Where $P(m_i \mid m_{i-1})$ is computed from the corpus.

Given below are few examples where bi-gram produces better results than unigram and maximum matching.

<table>
<tr><td>Input</td><td>لوگ پولیس کےڈنڈے کھانےکے بعد بھی بازنہیں آرہے ہیں۔</td></tr>
<tr><td>Max Matching</td><td>لوگ|پولیس|کے|ڈنڈے|کھانے|کے|بعد|بھی|باز|نہیں|آر|ہے|ہیں|</td></tr>
<tr><td>Unigram</td><td>لوگ|پولیس|کے|ڈنڈے|کھانے|کے|بعد|بھی|باز|نہیں|آر|ہے|ہیں|</td></tr>
<tr><td>Bigram</td><td>لوگ|پولیس|کے|ڈنڈے|کھانے|کے|بعد|بھی|باز|نہیں|آ|رہے|ہیں|</td></tr>
</table>

Figure 84: Max Matching Vs Unigrgram Vs Bigram Results

<table>
<tr><td>Input</td><td>میں جوبہکا تو میری</td></tr>
<tr><td>Max Matching</td><td>میں|جوب|ہکا|تو|میری|</td></tr>
<tr><td>Unigram</td><td>میں|جو|بہ|کا|تو|میری|</td></tr>
<tr><td>Bigram</td><td>میں|جو|بہکا|تو|میری|</td></tr>
</table>

Figure 85: Max Matching Vs Unigrgram Vs Bigram Results

Bigram will obviously do better then unigram because they consider context. However in order to obtain better results we need a huge training data of bigrams. In this work roughly 39,000 bigrams were used. These are two few to have any impact but the results are still comparable with unigram statistics as we will see.

7.3.11. Annotating Segmentations

This module is not directly linked with segmentation process. The idea is to annotate the output at different levels based on the phenomenon that we have discussed. Free morphemes are tagged as root, suffix, prefix or word. Root + affixes are tagged as word. Compounds, reduplications and abbreviations are tagged with their respective tagged. Nested tagging is done in this case because all of these are made up of words so their constituents are tagged a words. Given below are the tags for each of the phenomenon:

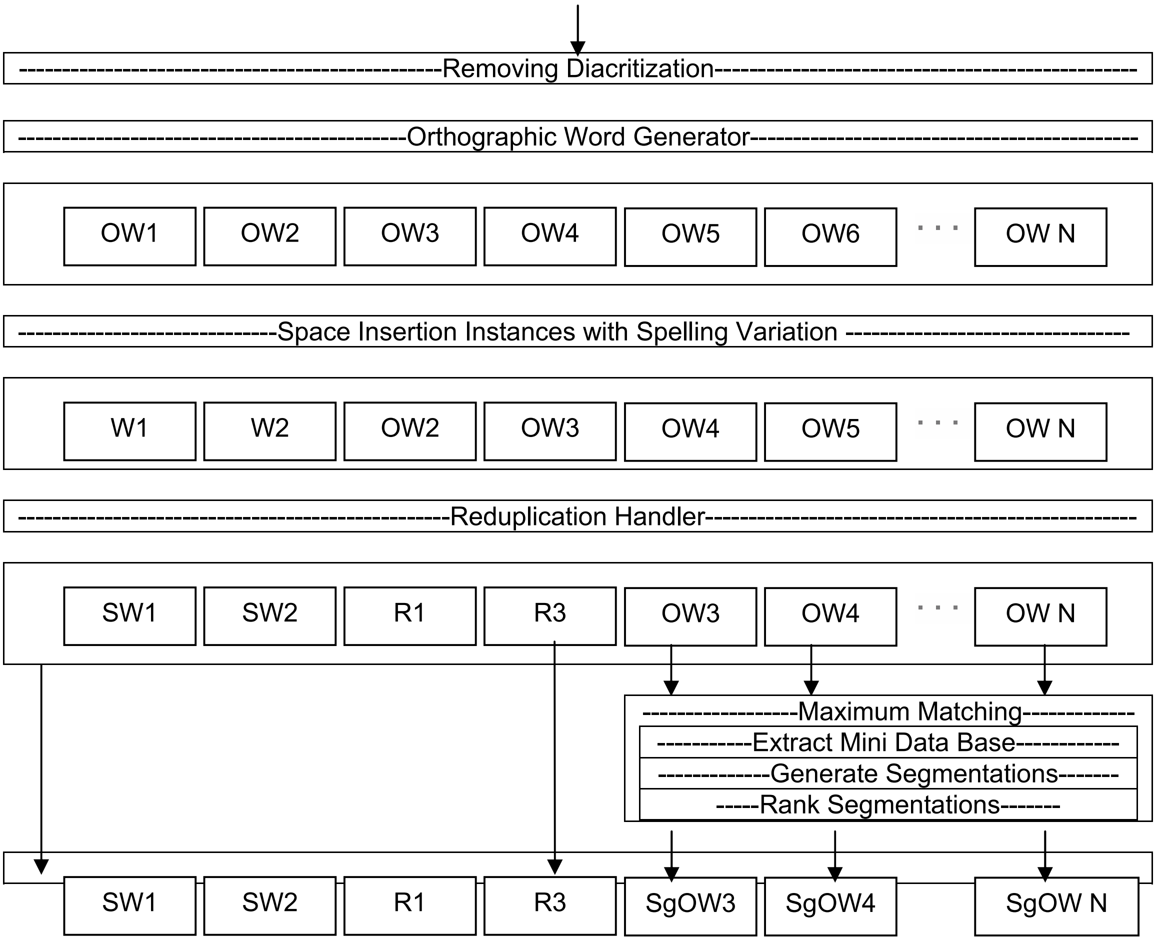
Table 11: Tags Used for Annotation + Examples

Phenomenon	Tags	Examples
Word	<W></W>	<W>اعلان</W>
Root	<R></R>	<W><R>ضرورت</R><S>مند</S><W>
Suffix	<S></S>	<W><R>حیرت</R><S>انگیز</S><W>
Prefix	<P></P>	<W><P>بد</P><R>تہذیبی</R><W>
XY Compounds	<C1></C1>	<C1><W>انشاء</W><W>اللہ</W><C1>
X-e-Y Compounds	<C2></C2>	<C2><W>وزیر</W><W>اعلی</W><C2>
X-o-Y Compounds	<C3></C3>	<C3><W>گرد</W><W>و</W><W>نواح</W><C3>
Reduplication	<Rd></Rd>	<Rd><W>ٹھیک</W><W>ٹھاک</W></Rd>
Abbreviations	<A></A>	<A><W>پی</W> <W>سی</W> <W>بی</W></A>

The nesting in these tags somewhat map on the model drawn in section 4 with morphemes (Root <R> Suffix <S> and Prefix <P>) at lowest level, words <W> at a level above them and finally compounds <C1><C2><C3>, Abbreviations <A> and Reduplication at a level higher than words.

7.4. Main Model

This section explains the main model, the algorithm as it works and how the components discussed above combine together to produce a final output. The algorithm only refers to the module names. The internel working of each module has already been discussed. The figure given below is the over all picture:



--Removing Diacritization--
--Orthographic Word Generator--
OW1
OW2
OW3
OW4
OW5
OW6
. . .
OW N
-----------------------------Space Insertion Instances with Spelling Variation --------------------------------
W1
W2
OW2
OW3
OW4
OW5
. . .
OW N
---Reduplication Handler---
SW1
SW2
R1
R3
OW3
OW4
. . .
OW N
------------------Maximum Matching-------------
-----------Extract Mini Data Base------------
-------------Generate Segmentations-------
-----Rank Segmentations-------
SW1
SW2
R1
R3
SgOW3
SgOW4
SgOW N

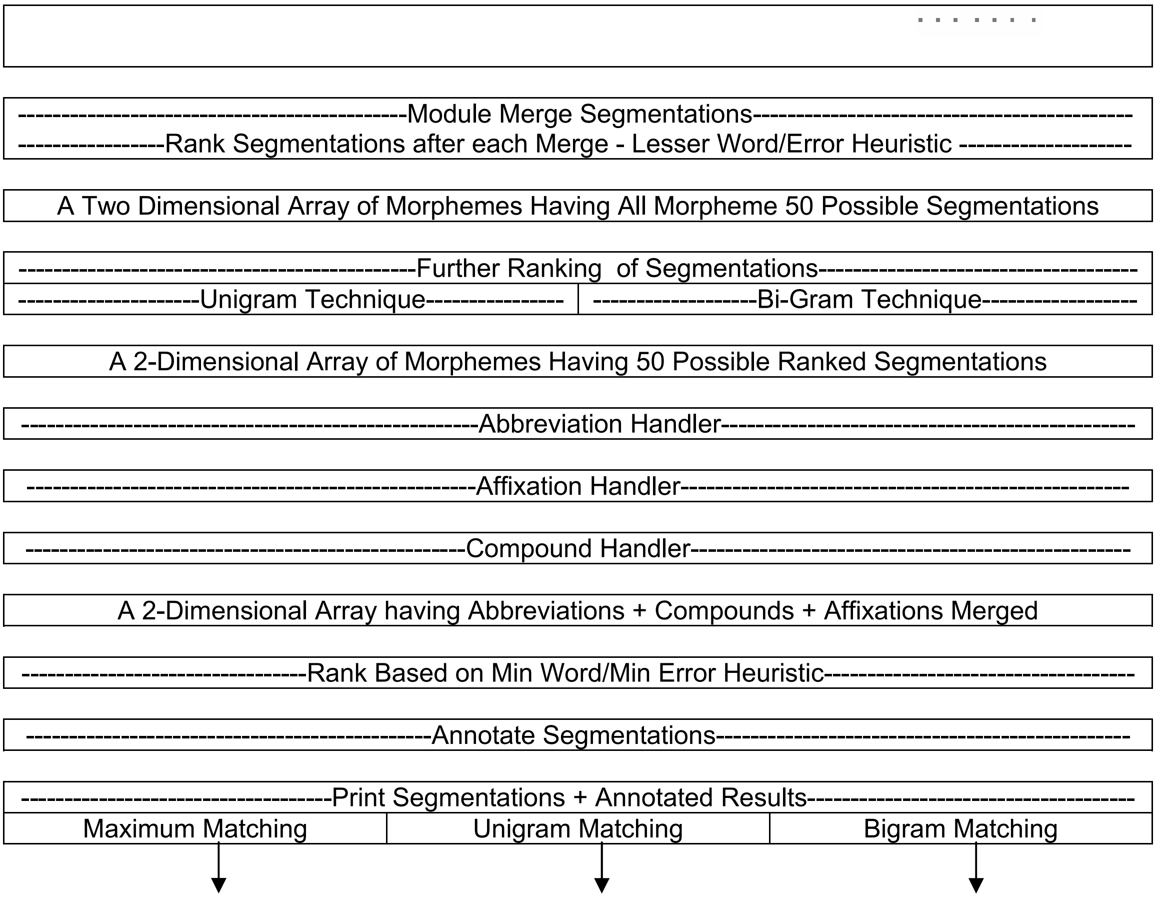

. . . .
Module Merge Segmentations
Rank Segmentations after each Merge - Lesser Word/Error Heuristic
A Two Dimensional Array of Morphemes Having All Morpheme 50 Possible Segmentations
Further Ranking of Segmentations
Unigram Technique
Bi-Gram Technique
A 2-Dimensional Array of Morphemes Having 50 Possible Ranked Segmentations
Abbreviation Handler
Affixation Handler
Compound Handler
A 2-Dimensional Array having Abbreviations + Compounds + Affixations Merged
Rank Based on Min Word/Min Error Heuristic
Annotate Segmentations
Print Segmentations + Annotated Results
Maximum Matching
Unigram Matching
Bigram Matching

Figure 86: The Overall Model

The architectural diagram is shown below:

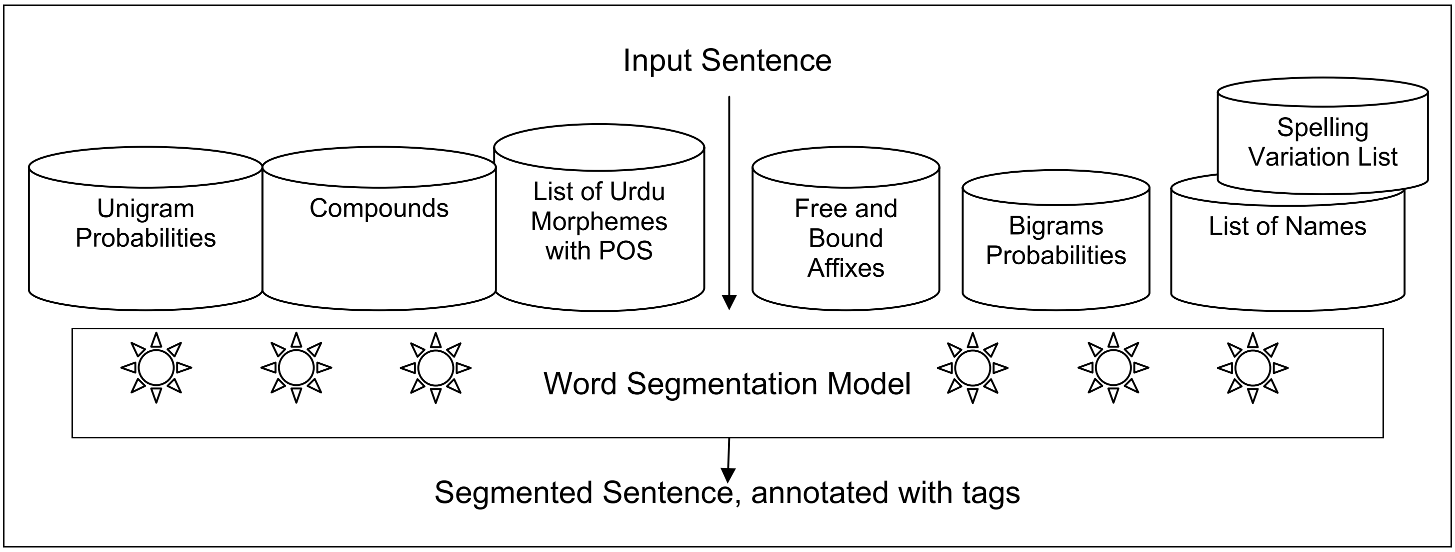


Figure 87: Architectural Diagram

7.4.1. Algorithm

Following are the steps taken to segment input sequence:

- Diacritics are removed from the input.
- Input is divided into orthographic words.
- Each orthographic word is send to “space insertion instance with spelling variations” module. This module might break some of the orthographic words into proper words.
- Each orthographic word is send to reduplication module.This module might break some orthographic words into its reduplication constituents or might identify two orthographic words as a reduplication instance.
- All orthographic words are sent to maximum matching module one by one. Maximum matching module sends back 10 best segmentations of an orthographic word if segmentations are more than 10. Ranking is done by min-word min-error heuristic.
- Segmentations are merged with other segmentations and the words that are identified through reduplication and spelling variation modules one by one. After each merge segementations are cut off to 50 segmentations selecting best 50 through min-word min-error heuristic. We now have 2-Dimensional array of top 50 segmentations where each cell in a 2-D array has a morpheme, reduplication word or word identified through spelling variation module. Each row in this 2-D array represents a possible segmentation.
- These segmentations are sent to unigram module. Unigram module extracts unigram probabilities of each of the morpheme in 2-D array, assigns an already decided unknown probability to unknown words. It then multiplies the probabilities of each morpheme in a row and finds out which row has maximum probability. This index is saved.
- These segmentations are then sent to bigram module. Bigram module extracts bigram probabilities of each of the consecutive morpheme pair in a row, assigns an already decided unknown probability to unknown pairs. It then multiplies the probabilities of each pair in the row and finds out which row has maximum probability. This index is saved.
- The 2-D array is sent to Abbreviation handler. Abbreviation handler merges some of the morphemes by identifying them as letters.
- The 2-D array is then sent to Affixation Handler. Affixation handler merges some of the morphemes by identifying prefixes and suffixes.
- The 2-D array is then sent to Compound Handler. The compound handler merges some of the morphemes by plain lookups from compound list.
- Now we have a 2-D array having abbreviations, affixation, compounds and reduplications merged together.
- Because some of the morphemes have been merged. We again rank our 50 segmentations on base of min-word min-heuristic.
- The algorithm then annotates segmentations

- 3 best segmentations based on maximum matching heuristic, unigram and bigram results are printed.

- Annotated forms of these are also printed.

The algorithm has been designed in layers. Each layer handles a particular phenomenon. The algorithm can skip any layer. It can give output at all the levels discussed in section 4. It can print output at morpheme level, at word level. It can go beyond this level and print at third level. Even at third level program can skip any layer and print compounds only, reduplications only and likewise.

8. Results

The algorithm was tested on a corpus of 2367 words. Word here means every thing (affixation, compounds, abbreviations, reduplication are also included). The corpus we selected contained 404 segmentation errors with 221 cases of space insertion problems and 183 cases of space deletion problems. In space deletion there were 66 cases of affixation, 63 cases of compounding, 32 cases of reduplication and 22 cases of abbreviations. The results for all three techniques are shown below:

Table 12 Percentage of Correctly Detected Words

	Correctly Detected Words	%age
Maximum Matching	2209/2367	93.3%
Unigram Technique	2269//2367	95.8%
Bigram Technique	2266//2367	95.7%

The statistical based unigram and Bigram clearly outperform maximum matching method. This is because maximum matching technique is a plain technique and does not use linguistic evidence. As compared to Unigram and Bigram techniques represent Urdu as they are extracted from corpra. Very few bigrams were used in this work. The results from Bigram technique are expected to improve a lot once the number of bigrams improves.

Table 13: Percentage of Number of Errors Detected

	Correctly Detected Errors	%age
Maximum Matching	323/404	79.95%
Unigram Technique	347/404	85.8%
Bigram Technique	339/404	83.9%

Table 14: Percentage of Number of Errors Detected Space Insertion and Deletion Breakage

	Space Insertion	%age	Space Deletion	%age
Maximum Matching	186/221	84.16%	132/183	72.13%
Unigram Technique	214/221	96.83%	133/183	72.67%
Bigram Technique	209/221	94.5%	130/183	71.03%

The results figure in space deletion problem is damaged by compounding problem. This is because 44.4 % compounds were successfully detected. Compounding is a very productive phenomenon in Urdu and obviously it is impossible to list all the compounds. If we remove compounding from space delection problems then we have successfully solved 105/120 problems i.e.87.5% which is reasonable.

Table 15: Percentage Breakage for Space Deletion Problem

	Affix	%age	Comp	%age	Redup	%age	Abbr	%age
Maximum	58/66	87.87%	28/63	44.44%	27/32	84.37%	19/22	86.36%
Unigram	59/66	89.39%	28/63	44.44%	27/32	84.37%	19/22	86.36%
Bigram	56/66	84.84%	28/63	44.44%	27/32	84.37%	19/22	86.36%

Errors in reduplication occurred because of the threshold was kept as 4. Some of the reduplication cases in which length of X and Y was 3 were not detected. There were 1-2 cases in which the edit distance was more than 1. These also contributed to error stats.

Error in abbreviation cases occurred because of the X X X name pattern which occurred as an exception. In the test corpus no such instance was found, the abbreviations consitently followed X X name format. Other errors occurred because of some English letters that are also valid words in Urdu.

The poor detection rate of compounds i.e. 44.44% does not represent the problem with the technique but with the list used to detect compounds. Only a list of 1800 compounds was used which a very little number is considering that compounding is a very rich phenomenon in Urdu. This also contributes to the fact that compounding is a problem higher than word segmentation and should be solved as a different problem on top of word segmentation layer.

If compounds, reduplication and abbreviations are not considered as words and we use the definition of word defined in section 4 then results differ. In this case the test corpus contains 2569 words with 287 segmentation errors with 221 space insertion problems and 66 space deletion problems (affixation problems). Given below are results in this scenario.

Table 16: Percentage of Correctly Detected Words

	Correctly Detected Words	%age
Maximum Matching	2454/2569	95.5 %
Unigram Technique	2514/2569	97.85%
Bigram Technique	2511/2569	97.77%

The percentage of detection is improved because the complexity of the problem has been reduced.

Table 17: Percentage of Number of Errors Detected

	Correctly Detected Errors	%age
Maximum Matching	244/287	85.01%
Unigram Technique	273/287	95.12%
Bigram Technique	265/287	92.33%

Table 18: Percentage of Number of Errors Detected Space Insertion and Deletion Breakage

	Space Insertion	%age	Space Deletion	%age
Maximum Matching	186/221	84.16%	58/66	87.87%
Unigram Technique	214/221	96.83%	59/66	89.39%
Bigram Technique	209/221	94.5%	56/66	84.84%

Space deletion problem in table 18 means only affixation. The errors found in all the cases attribute to the affixes like 'ان' and 'بر' and 'کار' affixes that are also free morphemes. The technique used in this work does not effectively the morphemes that occur frequently as free morphemes.

As can be seen Unigram technique was best of the lot for the data used in this thesis. An extended testing was done to test the results on a corpus of 64,883 words using Unigram technique. These results were obtained by keeping in mind the the definition of word in section 4, as defined within the scope of this thesis. Out of 64,883, 61529 words were correctly identified hence giving 94.83% results. The accuracy drop down, as compared to initial results attributes to the spelling mistakes in the input corpus. The initial nput corupus of 2500 was cleaned first. The results are likely to improve if same is done with this corpus.

9. Conclusion

This thesis presents a preliminary effort on word segmentation problem in Urdu. It is a multi-demensional problem. Each dimension requires a deeper study and analysis. Each sub-problem has been touched in this work and a basic solution for all has been devised. However to improve on results each of these modules require a separate analysis and study and hence a separate solution. Urdu has a different case than south East Asian languages where space insertion is the only problem. Urdu has no spaces and extra spaces. Both problems have their own dimensions and intracies trying to attack all simultaneously and bring one effective solution is very difficult. Doing so many things adversely affects efficiency. This work has effectively solved space insertion problem. However space deletion problem requires more study and analysis. As can be seen from results, the solution provided for compound is almost a non-solution. A deeper analysis of abbreviations and their patterns in Urdu are required. Initial corpus study did not revealed many instances of reduplications. Deeper analysis on reduplication patterns is required.
From the results it appears that unigram is better than bigram but that is not the case. Bigram technique has done much with too little information. For bigram technique to be affective huge amount of data is required.

10. Future Work and Improvements

As already said every dimension in this problem requires a deeper analysis and detailed study of patterns. Rule based approach is used for all the space deletion problems. Statistics is only used in ranking of segmentations. In future work statistics and bigram analysis can be used to merge morphemes. For example analysis such as whether, bigram probability of ‘پر|سکون’ is higher than unigram probability of 'پرسکون’ will be helpful to decide whether a morpheme is a word or affix. More corpus can be tagged to find out joining statistics of all the affixes (that can occur as free morpheme) can be dected. Such analysis will reveal whether an affix is more inclined towards joining or occurs freely more frequently.

Similarly a corpus can be tagged on compounds. For each morpheme its probability to occur in compound can be calculated. If two or more morphemes with higher compounding probabilities co-occur they can be joined together. Similar corpuses can be tagged for abbreviations.

Ranking of segmentations and affix merging can be improved if POS tags are also involved with bigram probabilities. Use of POS tags with n-gram technique is proven to be very helpful in solving unknown problems. Much more work has to be done in Urdu word segmentation. We have just stepped into it.

Reference:

1. Suhab, K. 2004. *Nigaristan.* Lahore: Dar-ul-tazkeer.

2. Platts, John T 1909. *A Grammar of the Hindustani or Urdu Language.* London: Crosby Lockwood and Son.

3. Schmidt, Ruth L. 1999. *Urdu, An Essential Grammar.* London: Routeledge Taylor & Francis Group.

4. Sproat, R., Shih, C. Gale, W. and Chang, N. 1996. A Stochastic Finite-State Word-Segmentation Algorithm for Chinese In *Computational Linguistics, Volume 22,Number 3*

5. Sornlertlamvanich, V., Potipiti, T., Wutiwiwatchai, C. and Mittrapiyanuruk, P. 2000. The State of Art in Thai Language Processing. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics.* Hong Kong

6. Javed, I. 1985. *Nai Urdu Qawaid,* Taraqqi Urdu Bureau, New Delhi.

7. O'Grady, Aronoff, M and Miller, R.2004. *Contemporary Linguistics - An Introduction:*Bedford/St. Martins.

8. Payne, Thomas E.2006. *Exploring Language Structure, A Student's Guide.* Cambridge: Cambridge University Press.

9. Sproat, R. 1992. *Morphology and Computation.* The MIT Press

10. Wikipedia, "Compound Linguistics," http://en.wikipedia.org/wiki/Compound_(linguistics)

11. Basic Concepts of Morphology [Systheo_S05_06_lecture7]

12. systheo_05_06_lecture7 (basic concepts of morphology)

13. Sabzwari, S. 2002, *Urdu Quwaid*. Lahore: Sang-e-Meel Publication

14. Albert Bickford. J, 1998. *Morphology and Syntax: Tools for Analyzing the World's Languages*: Summer Inst of Linguistics

15. Spelling and Compound Words http://www.rand.org/clients/creative_services/style_manual_ext/Style2-2.pdf

16. Poowarawan, Y., 1986. Dictionary-based Thai Syllable Separation. In *Proceedings of the Ninth Electronics Engineering Conference*

17. Rarunrom, S., 1991. Dictionary-based Thai Word Separation. Senior Project Report.

18. Durrani, N., Dalolay, V. 2005. Lao Line Breaking Algorithm. *PAN Localization Project*

19. Wong, P., Chan, C. 1996. Chinese Word Segmentation based on Maximum Matching and Word Binding Force. In *Proceedings of COLING 96*, pp. 200-203.

20. Sornlertlamvanich, V. 1995. Word Segmentation for Thai in a Machine Translation System (in Thai), *Papers on Natural Language Processing, NECTEC, Thailand*

21. Theeramunkong, T., Usanavasin, S. 2001. Non-Dictionary-Based Thai Word Segmentation Using Decision Trees. In *Proceedings of the First International Conference on Human Language Technology Research*, pages 00-00, San Diego, Californian

22. Tsai, C. 1996. MMSEG: A Word Identification System for Mandrin Chinese Text Based on Two Variants of the Maximum Matching Algorithm.

23. Yeh, C and Lee, H. 1991. Rule-based word identification for Mandarin Chinese sentences—a unification approach. *Computer Processing of Chinese and Oriental Languages*, 5(2):97-118.

24. Chen, K and Liu, S. 1992. Word identification for Mandarin Chinese sentences. In *Proceedings of COLING-92*, pages 101-107. COLING.

25. Liang, N. 1986. A written Chinese automatic segmentation system-CDWS. In *Journal of Chinese Information Processing*, 1(1):44-52.

26. Li, B.Y., S. Lin, C.F. Sun, and M.S. Sun. 1991. A maximum-matching word segmentation algorithm using corpus tags for disambiguation. In *ROCLING IV*, pages: 135-146, Taipei. ROCLING

27. Gu, P. and Mao, Y. 1994. The adjacent matching algorithm of Chinese automatic word segmentation and its implementation in the QHFY Chinese-English system. In *International Conference on Chinese Computing*, Singapore.

28. Nie, J., Jin W., and Hannan, M. 1994. A hybrid approach to unknown word detection and segmentation of Chinese. In *International Conference on Chinese Computing,* Singapore.

29. Aroonmanakul, W. 2002. Collocation and Thai Word Segmentation. In *proceeding of SNLPOriental* COCOSDA.

30. Krawtrakul, A., Thumkanon. C., Poovorawan. Y. and Suktarachan. M. 1997. Automatic Thai Unknown Word Recognition. In *Proceedings of the natural language Processing* Pacific Rim Symposium.

31. Pornprasertkul, A., 1994. Thai Syntactic Analysis. *Ph.D. Thesis,* Asian Institute of Technology

32. Meknawin, S. 1995. Towards 99.99% Accuracy of Thai Word Segmentation. Oral Presentation *at the Symposium on Natural Language Processing in Thailand'95*

33. Kawtrakul, A. Kumtanode, S. 1995. A Lexibase Model for Writing Production Assistant System. In *Proceedings of Symposium Natural Language Processing in Thailand'95*

34. Theeramunkon, T. Sornlertlamvanich, V., Tanhermhong T., Chinnan, W. 2000. Character-Cluster Based Thai Information Retrieval, In *the Proceedings of Fifth International Workshop on Information Retrieval with Asian Languages.* Honkong, pp 75-80

35. Matsukawa, T., Miller, S., and Weischedel, R. 1993. Example-Based Correction of Word Segmentation and Part of Speech Labeling. *Human Language Technology*, p. 227-232.

36. Papageorgiou, C. 1994. Japanese Word Segmentation by Hidden Markov Model. In *Proceedings of the Human Language Technologies Workshop (HLT)*, pages 283-288.

37. Charoenpornsawat, P., Kijsirikul, B. 1998. Feature-Based Thai Unknown Word Boundary Identification Using Winnow. In *Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS'98)*.

38. Meknavin. S., Charenpornsawat. P. and Kijsirikul. B. 1997. Feature-based Thai Words Segmentation. NLPRS, Incorporating SNLP.

39. Aroonmanakun, W. 2002. Collocation and Thai Word Segmentation. In Thanaruk Theeramunkong and Virach Sornlertlamvanich, eds. *Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop*. Pathumthani: Sirindhorn International Institute of Technology. 68-75.

40. Blum, A. 1997. Empirical Support for Winnow and Weighted-Majority Algorithm: Results on a Calendar Scheduling Domain, Machine Learning, 26:5-23.

41. What is Word?
http://www.sussex.ac.uk/linguistics/documents/essay_-_what_is_a_word.pdf

42. Jurafsky, D. and Martin, J. H., Speech and Language Processing.: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition Prentice Hall; 1st edition (January 26, 2000)

Appendix A

Table 19: Oblique Pronouns with ‘کو’ Construction

With and Without Space		Alternative Construction	Meaning
مجھ کو	مجھکو	مجھے	To me
تجھ کو	تجھکو	تجھے	To You
جس کو	جسکو	جسے	To the one
جن کو	جنکو	جنھیں	To the ones
کس کو	کسکو	کسے	To whom
کن کو	کنکو	کنھیں	To whom
اس کو	اسکو	اسے	To him
ان کو	انکو	انھیں	To them
اپ کو	اپکو	-	To you
ہم کو	ہمکو	ہمیں	To us

Table 20: Possessive Pronouns with ‘کا’ Construction

With and Without Space		Meaning
اپ کا	اپکا	Yours
اپ کی	اپکی	Yours
اپ کے	اپکے	Yours
ان کا	انکا	Theirs
ان کی	انکی	Theirs
ان کے	انکے	Theirs
جس کا	جسکا	The one whose
جس کی	جسکی	The one whose
جس کے	جسکے	The one whose
اس کا	اسکا	His/Hers
اس کی	اس کی	His/Hers
اس کے	اسکے	His/Hers
کن کا	کنکا	Whose
کن کی	کنکی	Whose
کن کے	کنکے	Whose
تس کا	تسکا	His/Hers
تس کی	تسکی	His/Hers
تس کے	تسکے	His/Hers
تن کا	تنکا	Theirs
تن کی	تنکی	Theirs
تن کے	تنکے	Theirs

Appendix B

Table 21: Adverbs of Time and Manner

With and Without Space		Meaning
اس وقت	اسوقت	This/that time
کس وقت	کسوقت	Which time
جس وقت	جسوقت	Whenever
اس طرف	اسطرف	This direction
کس طرف	کسطرف	Which direction
جس طرف	جسطرف	Wherever
اس طرح	اسطرح	This way
کس طرح	کسطرح	Which way
جس طرح	جسطرح	Whatever way
یہاں پر	یہانپر	Over Here

Appendix C

Table 22: Postpositional Phrases

With and Without Space		Meaning
کی طرف	کیطرف	Direction of
کی وجہ	کیوجہ	Because of
کی طرح	کیطرح	Manner of
کے خلاف	کیخلاف	Against
کے لیے	کیلیے	For

Appendix D

Table 23: Compound Verbs Joiners

With and Without Space		Meaning
دے دیا	دیدیا	Given
دے دی	دیدی	Given
جائے گا	جائیگا	Will go
جائے گی	جائیگی	Will go
کرے گا	کریگا	Will do
کرے گی	کریگی	Will do
لے کر	لیکر	After taking

Appendix E

Table 24: Suffixation

Suffixation	Meaning
ذمہ دار	Responsible
رشتہ داروں	Relatives
یقین دہانی	Assured
جاری کردہ	Issued
زلزلہ زدگان	Earth-quake victims
راہ گیر	Passenger
منظور کردہ	Approved
خود ساختہ	Voluntary
من پسند	Favorite
مفاد پرست	Selfish
جلد بازی	Haste
مغرب زدہ	Westernized
غیر مسلم	Non-Muslim
دہشت گردی	Terrorism
درخواست گزار	Applicant
قرآن خوانی	Quran Khuwani
دہشت گردوں	Terrorists
معذرت خواہ	Sorry
متاثر کن	Effective
خوب صورتی	Beauty
ذمہ داران	Responsible
پیش رفت	Progress
منصوبہ بندی	Planning
بندر گاہوں	Seaports
بے لوث	Selfless
بے داغ	Spotless
بااعتماد	Trustworthy
سرمایہ کاری	Finance
تیار کردہ	Prepared
غیر ملکی	Foreign

Appendix F

Table 25: Sample Data for Survey

- ❖ آج ملک بھر میں جشن آزادی انتہائی جوش و خروش اور عقیدت مندی سے منایا گیا۔ آزادی کا یہ پچاسواں سال ہم سب کیلیے ایک سنگ میل ہے۔

- ❖ گزشتہ روز صدرجنرل پرویزمشرف نے وزیرمملکت شوکت عزیز کے ہمراہ زلزلہ زدہ علاقوں کا دورہ کیا۔وزیراعلی پنجاب چوہدری پرویزالہی بھی ان کے ہمراہ تھے۔(شعبۂ نشرواشاعت)

- ❖ میچ کی صورتحال کافی سنگین ہے۔خدانخواستہ ہارجانے کی صورت میں پاکستان ٹورنمنٹ سے باہر کردیاجائیگا۔دور دراز علاقوں سے ائےہوے شائقین کے لیے یہ بات ناقابل برداشت ہوگی۔

- ❖ اج اہلیان پاکستان کا غیص وغصب قابل دید ہے۔ ملک میں امن واماں اور نظم وضبط قائم رکھنا محال ہے۔شہربھر میں جگہ جگہ ریلیاں نکالی جارہی ہیں۔ طلبہ وطلبات کارکنان ورضاکاران علماءومشائخ سبھی ان ریلیوں میں شامل ہیں۔ بگڑتی ہوئ صورت حال کو مدنظر رکھتے ہوئے پولیس ورینجرز کونئے احکامات جاری کردیے گئے ہیں جن پر جلد عمل درامد شروع ہوجائے گا (نوائےوقت)۔

- ❖ اس سال حج وعمره کی سہولیات کو مزید بہتربنایاجارہاہے۔اس سلسلےمیں خادمین حرمین شریفین کی کاوشیں قابل تعریف ہیں۔

- ❖ آج غلام اسحاق خان صاحب کو سپردخاک کردیاجائیگا ۔انکانمازجنازہ صبح دس بجے ادا ہوگا۔ صدرمشرف اور وزیراعظم شوکت عزیز نے انکے خاندان اور عزیزواقارب کیساتھ اظہار تعزیت و ہمدردی کیا۔ حکومت پاکستان انکی خدمات کو سرہاتی ہے اور انہیں خراج تحسین پیش کرتی ہے۔

- ❖ قائم مقام صدر نے فیصلہ کیا ہے کہ جیلوں میں پیدا ہونے والے جرائم پیشہ خواتین کے بچوں کو سرکاری سکولوں میں مفت تعلیم مہیا کی جائے۔تاکہ انہیں بھی اچھی تعلیم و تربیت اور خشگوار مستقبل میسر اسکے۔ یہ وزارت عظمی کا ایک انتہائ خوش آئندقدم ہے۔

- ❖ طالب علموں کو تعلیم کیساتھ کھیلوں اور دیگر سرگرمیوں کو بھی مناسب وقت دینا چاہیے۔ذہنی افزائش کے ساتھ ساتھ جسمانی نشو ونما بےحد ضروری ہے۔

30 subjects from different fields were given these paragraphs to mark word boundaries. These included computer scientists, linguists, computational linguists, Urdu and Farsi teachers and laymen. Results are shown in tables below where 1, 2 and 3 mentioned in table below tells reports the tally on that particular word.

Table 26: Compound Words with Linking Morpheme –e–

Problem Word	1	2	3
جشن آزادی	25	5	X
سنگ میل	27	3	X
وزیرمملکت	24	6	X
وزیراعلی پنجاب	0	27	3
شعبہ ٔ نشرواشاعت	18	8	4
صورتحال	28	2	X
صورت حال	28	2	X
ناقابل برداشت	24	5	1
اہلیان پاکستان	18	12	X
نوائےوقت	29	1	X
خادمین حرمین شریفین	14	5	10
قابل تعریف	22	8	X
قابل دید	23	7	X
سپردخاک	26	4	X
نمازجنازہ	22	8	X
وزیراعظم	28	2	X
حکومت پاکستان	17	13	X
خراج تحسین	23	7	X
وزارت عظمی	28	2	X

Table 27: Compound Words with Linking Morpheme –o–

Problem Word	1	2	3
جوش و خروش	27	3	X
نشرواشاعت	26	4	X
امن وامان	25	5	X
نظم وضبط	25	5	X
طلبہ وطلبات	21	9	X
کارکنان ورضاکاران	17	13	X
علماءومشائخ	21	9	X
پولیس ورینجرز	17	13	X
عزیزواقارب	25	5	X
اظہار تعزیت و ہمدردی	8	12 (5+7)[7]	10
تعلیم و تربیت	13	7	X
نشو ونما	29	1	X

Table 28: Derivational Suffixation, XY Compounding and Reduplication

Problem Word	1	2	3
عقیدت مندی	30	0	X
زلزلہ زدہ	27	3	X
دور دراز	25	5	X
جگہ جگہ	3	27	X
ساتھ ساتھ	3	27	X
عمل درامد	25	5	X
قائم مقام	23	7	X
جرائم پیشہ	22	8	X
خوش آئند	25	5	X
بےحد	28	2	X

Table 29: Post Positions, Compound Verbs and Personal Pronouns

Problem Word	1	2
کیلیے	20	10
کے لیے	1	29
ہارجانے	2	28
ائےہوے	0	30
ہو گی	15	15
ہوجائے	2	28
کردیا	9	21
جائیگا	18	12
جائے گا	5	25
کیساتھ	14	11
کے ساتھ	0	30
انکی	20	10
آسکے	16	14

Appendix G

Table 30: Free and Bound Prefixes with Status

Free	Status of Free Morphemes	Bound Prefixes
از	NOM/ADJ/ADV	بہر
ان	HAR/VER/NOM/	تہہ
با	NOM/ADJ/	ری
باز	NOM/	سوڈو
بد	NOM/ADJ	
برائے	NOM	
بن	NOM	
بیش	NOM/VER	
بے	NOM/ADJ/VER/AV	

[7] اظہار تعزیت and ہمدردی joined with linked morpheme –و– or اظہار and تعزیت و ہمدردی

پا	NOM	
پائے	NOM	
پر	NOM/ADJ	
پس	NOM	
پیش	ADJ/NOM	
خرد	NOM	
خود	NOM/ADJ	
خوش	NOM/ADJ	
در	NOM/ADJ	
دریں	NOM	
زبر	NOM	
زود	NOM	
زیر	NOM	
سر	NOM/ADJ	
سہ	NOM/ADJ	
صاحب	NOM	
صد	NOM	
غیر	NOM/ADJ/ADV	
فرو	NOM	
گل	NOM	
لا	NOM/ADJ	
مہا	NOM/ADJ	
نا	NOM/ADJ/ADV	
ہشت	NOM	
ہفت	NOM	
ہم	NOM/ADJ	
ہمہ	NOM/ADJ	
یک	NOM/ADJ	
الٹرا	NOM/ADJ	
انٹر	NOM/ADJ	
انڈر	NOM/ADJ	
اوور	NOM/ADJ	
ایکس	NOM	
ایکسٹرا	NOM/ADJ	
اینٹی	NOM/ADJ	
آؤٹ	NOM/ADJ	
آٹو	NOM	
پرو	NOM/ADJ	
پری	NOM/ADJ	
پوسٹ	NOM	
پولی	NOM/ADJ	
سپر	NOM	
ملٹی	NOM/ADJ	
منی	NOM	
ٹیلی	NOM	
بال	NOM/ADJ	
بلا	NOM/ADV	

Table 31: Free and Bound Suffixes with Status

Free Suffixes	Status	Bound Suffixes
ازار	NOM	افزائی
افروز	NOM	اندوز
افزا	NOM	اندوزوں
افشاں	NOM	اندوزی
انداز	NOM	اندیشانہ
اندازی	NOM	اندیش
اندام	NOM/ADJ	اندیشی
آرا	NOM	انگیز
آزار	NOM	انگیزی
آزاری	NOM	انگیزیوں
آزما	NOM	آرائی
آزمائی	NOM	آگیں
آشام	NOM	آمیز
آفریں	NOM	آور
آلودہ	NOM	آوروں
آموز	NOM	آوری
آمیزی	NOM/ADJ	آہنگی
آویز	NOM	بختی
باختہ	NOM	بخود
بازوں	NOM/ADJ	بدوش
باز	NOM/ADJ	بدوشوں
بازی	NOM/ADJ	برداری
باش	NOM/ADJ	بیں
باف	NOM	پرستی
بخش	NOM	پروری
بردار	NOM	پزیری
بند	NOM	پزیر
بندی	NOM	پسندی
بوسی	NOM	پسندانہ
پاشی	NOM	پوشوں
پرست	NOM	پیما
پرور	NOM	تی
پسندوں	NOM	چاری
پسند	NOM	چی
پن	NOM/ADJ	خواروں
پوش	NOM/ADJ	خوارگی
جات	NOM	خوانی
حال	ADJ	خواں
خاطر	NOM	خوراکی
خانے	NOM/ADJ	خوفی
خانہ	NOM/ADJ	خیز
خوار	NOM	خیزی
خواہ	NOM/ADJ	دارنی
خور	NOM	داران
خوری	NOM/ADJ	دارانہ
دادوں	NOM	داریوں
دارہ	NOM	داز

دار	NOM	دازی
داروں	NOM	دستیوں
داری	NOM	دلانہ
دانوں	NOM/ADJ	دوز
دان	NOM/ADJ	راں
دانی	NOM	زادے
دستی	NOM	زدگان
دستوں	NOM/ADJ	زدگی
دست	NOM/ADJ	زدوں
دل	NOM/ADJ	سازی
دلی	NOM/ADJ	شدگان
دہ	NOM	شکنی
دہانی	NOM	شماری
دہی	NOM	غرضی
رنگی	NOM/NUM	فروشوں
ریز	NOM	فشانی
ریزی	NOM	فشاں
زاد	NOM	فگن
زادہ	NOM	فیہ
زادی	NOM	قسمتی
زدہ	NOM/ADJ	کاران
زن	NOM	کارانہ
زنی	NOM	کاریاں
ساز	NOM	کدوں
ستان	NOM/ADJ	کشوں
ستانی	NOM/ADJ	کشی
سرا	NOM	کناں
سرائی	NOM	کنندگان
سنج	NOM	گاریوں
سوز	NOM	گاران
سوزی	NOM	گار
کردہ	NOM/ADJ	گاری
شدہ	NOM	گاہیں
شعار	NOM	گردی
شکن	NOM	گسار
شناس	NOM	گوئی
صورت	NOM/ADJ	گیں
طراز	NOM	گہ
طرازی	NOM	گیروں
فرسا	NOM	گیں
فروش	NOM	مندانہ
فروشی	NOM	مندوں
فگار	NOM/ADJ	ناکی
فہم	NOM/ADJ	نشینی
فہمی	NOM/ADJ	نگیں
کار	NOM/ADJ/ADV	نماں
کاری	NOM/ADJ/ADV	نوردی
کدہ	NOM	نوشوں
کدے	NOM	نویسی
کرام	NOM	واریت

داروں	NOM	دستیوں
داری	NOM	دلانہ
دانوں	NOM/ADJ	دوز
دان	NOM/ADJ	راں
دانی	NOM	زادے
دستی	NOM	زدگان
دستوں	NOM/ADJ	زدگی
دست	NOM/ADJ	زدوں
دل	NOM/ADJ	سازی
دلی	NOM/ADJ	شدگان
دہ	NOM	شکنی
دہانی	NOM	شماری
دہی	NOM	غرضی
رنگی	NOM/NUM	فروشوں
ریز	NOM	فشانی
ریزی	NOM	فشاں
زاد	NOM	فگن
زادہ	NOM	فیہ
زادی	NOM	قسمتی
زدہ	NOM/ADJ	کاران
زن	NOM	کارانہ
زنی	NOM	کاریاں
ساز	NOM	کدوں
ستان	NOM/ADJ	کشوں
ستانی	NOM/ADJ	کشی
سرا	NOM	کناں
سرائی	NOM	کنندگان
سنج	NOM	گاریوں
سوز	NOM	گاران
سوزی	NOM	گار
کردہ	NOM/ADJ	گاری
شدہ	NOM	گاہیں
شعار	NOM	گردی
شکن	NOM	گسار
شناس	NOM	گوئی
صورت	NOM/ADJ	گیں
طراز	NOM	گہ
طرازی	NOM	گیروں
فرسا	NOM	گیں
فروش	NOM	مندانہ
فروشی	NOM	مندوں
فگار	NOM/ADJ	ناکی
فہم	NOM/ADJ	نشینی
فہمی	NOM/ADJ	نگیں
کار	NOM/ADJ/ADV	نماں
کاری	NOM/ADJ/ADV	نوردی
کدہ	NOM	نوشوں
کدے	NOM	نویسی
کرام	NOM	واریت

خانوں	NOM/ADJ	پرستیوں
دادیں	NOM	خوانیاں
دانیاں	NOM	خوانیوں
دراز	NOM	خوروں
درازی	NOM	خوریوں
درازیاں	NOM	خیزوں
ریزوں	NOM	خیزیاں
سراؤں	NOM/ADJ/ADV	خیزیوں
سرائیوں	NOM/ADV/ADJ	داریاں
شعاری	NOM	دانیوں
فگاروں	NOM/ADJ	درازیوں
کاروں	NOM/ADJ	دبانیاں
مندیاں	NOM	دبانیوں
نامہ	NOM	رانیوں
ناموں	NOM	ریزیاں
نامیوں	NOM/ADJ	ریزیوں
نگاروں	NOM	زادوں
طراز	NOM	زادیاں
طرازی	NOM	زادیوں
طرازیاں	NOM	سازیاں
طرازیوں	NOM	سازیوں
بین	NOM	سرائیاں
بینوں	NOM	سگالیاں
تراش	NOM	شناسی
تراشی	NOM	ظرفیاں
گزاروں	NOM	ظرفیوں
گزاری	NOM	غرضیاں
گزار	NOM/ADJ	غرضیوں
گھر	NOM	فشانیاں
گھروں	NOM	فشانیوں
لیوا	NOM	فہمیاں
بانی	NOM	فہمیوں
نمائی	NOM/ADJ	کاریوں
نماؤں	NOM	کشائیاں
بندی	NOM	کشائیوں
بندیاں	NOM	گردیاں
بندیوں	NOM	گردیوں
گروں	NOM	گساروں
گران	NOM	گوئیاں
گری	NOM	گوئیوں
گر	NOM	گیریاں
گرنی	NOM	گیریوں
جوئی	NOM	لوحیاں
بازوں	NOM/ADJ	مزاجیاں
نامے	NOM	مزاجیوں
آباد	NOM/ADJ	مندیوں
آبادی	NOM	ناکیاں
گاروں	NOM/ADJ	ناکیوں
عامہ	NOM/ADJ	نامیاں
		نفسیاں

		خوانیاں
		خوانیوں
		خوروں
		خوریوں
		خیزوں
		خیزیاں
		خیزیوں
		داریاں
		دانیوں
		درازیوں
		دبانیاں
		دبانیوں
		رانیوں
		ریزیاں
		ریزیوں
		زادوں

Appendix H

```
Let s1[ ] be a character Array
Let s2 [ ] be a character Array
distance [ ][ ] be a 2-D integer Array

distance [ 0 ][ 0 ] = 0
distance [ i ] [ 0 ] = i for i=0….|s1|
distance [ j ][ 0 ] = j for j=0….|s2|

distance[ i ][ j ] = min(distance [ i-1 ] [ j-1 ]
+ X, distance[ i-1 ][ j ]+1, distance[ i ] [ j-1 ]
+1) where X=0 if s[ i ] = s[ j ] else 1
```

Figure 88: Single Edit Distance Algorithm